Combining Perceptions and Prescriptions in Requirements Engineering Process Assessment: An Industrial Case Study

Nannette P. Napier, Lars Mathiassen, *Member*, *IEEE*, and Roy D. Johnson

Abstract—Requirements engineering (RE) is a key discipline in software development and several methods are available to help assess and improve RE processes. However, these methods rely on prescriptive models of RE; they do not, like other disciplines within software engineering, draw directly on stakeholder perceptions and subjective judgments. Given this backdrop, we present an empirical study in RE process assessment. Our aim was to investigate how stakeholder perceptions and process prescriptions can be combined during assessments to effectively inform RE process improvement. We first describe existing methods for RE process assessment and the role played by stakeholder perceptions and subjective judgments in the software engineering and management literature. We then present a method that combines perceptions and prescriptions in RE assessments together with an industrial case study in which the method was applied and evaluated over a three-year period at *TelSoft*. The data suggest that the combined method led to a comprehensive and rich assessment and it helped *TelSoft* consider RE as an important and integral part of the broader engineering context. This, in turn, led to improvements that combined plan-driven and adaptive principles for RE. Overall, the combined method helped *TelSoft* move from Level 1 to Level 2 in RE maturity, and the employees perceived the resulting engineering practices to be improved. Based on these results, we suggest that software managers and researchers combine stakeholder perceptions and process prescriptions as one way to effectively balance the specificity, comparability, and accuracy of software process assessments.

Index Terms—Process implementation and change, qualitative process analysis, requirements engineering process, software management, software process.

1 Introduction

Requirements Engineering (RE) covers all aspects of the discovery, documentation, and maintenance of requirements throughout the software development life cycle (e.g., [1]). An expert panel consisting of both practitioners and academics agreed that the RE process remains the most problematic of all software engineering activities [2]. Furthermore, software project managers ranked the problem of misunderstood requirements as their second most important risk to be managed [3]. Many companies, therefore, seek to improve their RE processes by engaging in software process improvement (SPI) projects.

SPI projects are complex organizational change efforts in which it is necessary to take the particular context of processes, people, and values into account [4]. As a consequence, there is no single best way to assess and improve RE processes or software processes in general. Instead, software managers are challenged to select between different types of assessment and improvement methods to best suit their specific context and available resources [5], [6].

Methods for software process assessment and improvement can be classified into two main categories: *inductive* or problem-based and *prescriptive* or model-based [5], [7]. Inductive methods center improvement on the specific strengths and weaknesses of existing processes and consider stakeholders' perceptions as important knowledge sources. In contrast, prescriptive methods center improvement on the best practices within industry. While the dominant approaches to software process assessment and improvement are prescriptive (e.g., [8], [9]), there are a number of inductive approaches available (e.g., [5], [7], [10], [11], [12]) and stakeholder perceptions and subjective judgments have more generally been adopted as an important approach to inform other software disciplines (e.g., [13], [14], [15], [16]).

Three particular challenges are discussed in the literature that software managers should consider when selecting assessment methods.

- Assessment *specificity*: How well does the assessment uncover improvement problems and opportunities within the company? This challenge is typically addressed through inductive methods (e.g., [7], [10], [11], [12], [17]). Besides providing insight into the specific improvement context, these methods help achieve buy-in from stakeholders by involving them in the process.

- *N.P. Napier is with the School of Science and Technology, Georgia Gwinnett College, 1000 University Lane, Lawrenceville, GA 30043. E-mail: nnapier@ggc.edu.*
- *L. Mathiassen is with the Center for Process Innovation, J. Mack Robinson College of Business, Georgia State University, 35 Broad Street, NW, Suite 427, Atlanta, GA 30303. E-mail: lmathiassen@gsu.edu.*
- *R.D. Johnson is with the Informatics Department, Faculty of Engineering, Built Environment, and Information Technology, University of Pretoria, IT Building 5-95, Lynnwood Road, Hatfield, Pretoria, Gauteng 0002, South Africa. E-mail: roy@up.ac.za.*

Manuscript received 3 June 2008; revised 8 Dec. 2008; accepted 13 Jan. 2009; published online 15 May 2009.
Recommended for acceptance by N. Maiden.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number TSE-2008-06-0187.
Digital Object Identifier no. 10.1109/TSE.2009.33.

- Assessment *comparability*: How well does the assessment benchmark against the best practices within the industry? This challenge is typically addressed through prescriptive methods (e.g., [8], [9], [18], [19], [20]). In addition to supporting benchmarking, these methods help companies assess all key process areas.
- Assessment *accuracy*: How well does the assessment represent process characteristics within the company? This challenge is typically addressed through adoption of comprehensive data collection using multiple measures. In practice, assessment accuracy might be compromised because SPI efforts are constrained by limited resources [6], [7], [21], [22].

With this background in mind, the aim of this research was to investigate how to effectively balance the assessment challenges involved in RE process improvement. Although a number of focused RE assessment methods have been developed [23], [24], [25], [26], they all take a prescriptive approach. Such approaches would meet the assessment comparability challenge, but may fare poorly with respect to assessment specificity and accuracy. To this end, we developed an assessment approach that combines perceptions and prescriptions. Over a three-year period, the method was twice applied and evaluated within the software company *TelSoft* [27], [28].[1]

The resulting research offers two contributions to the software engineering body of knowledge. First, the research contributes to the RE assessment literature; existing methods are all prescriptive and the presented method is the first to draw systematically on stakeholder perceptions. Second, the research contributes to the general software process assessment literature; while there are a number of inductive methods that draw upon stakeholder perceptions (e.g., [7], [10], [11], [12], [17]), the presented method is the first that systematically combines stakeholder perceptions and process prescriptions to inform SPI.

In the following narrative, we present the background literature and describe the combined assessment method and the industrial case study. Based on the results from *TelSoft*, we discuss the contributions in relation to the existing literature on RE assessment and software process assessment in general. Finally, we conclude with implications for software research and practice.

2 Theoretical Background

In the following, we review current methods for RE process assessment and the role played by stakeholder perceptions and subjective judgment in software engineering and management disciplines.

2.1 Methods for RE Assessment

Researchers have used three main methods for RE process assessment: analyzing the RE-related data from generic software process assessments (e.g., SW-CMM or ISO/IEC 15504), applying an RE-specific version of the SW-CMM, and measuring adherence to the best practices based on a dedicated RE maturity model.

The first method relies on general models for software process assessment. For example, El Emam and Birk [29] used a subset of the assessment data from 44 organizations during the ISO/IEC 15504 trials [30] to examine whether the Software Requirements Analysis process capability is positively related to overall project performance. Damian et al. [24] similarly studied the benefits of RE process improvement using SW-CMM mini assessments.

The second method relies on RE-specific versions of general maturity models. Beecham et al. have developed an RE model based on the SW-CMM called R-CMM [23]. They apply the Goal-Question-Metric paradigm [31] and associate high-level RE goals with the different maturity levels ranging from initial (Level 1) to optimizing (Level 5). An example of a high-level goal to reach Level 2 is "to implement a repeatable RE process" [23]. Related to each goal is a set of assessment questions to ask about RE processes and their relation to the best practices. Weaknesses identified in the analysis are then used to suggest RE improvement goals.

The third method is uniquely focused on RE as suggested in the Requirements Engineering Good Practice Guide (REGPG) [26]. The REGPG describes 66 RE guidelines within eight areas—requirements documents, requirements elicitation, requirements analysis and negotiation, describing requirements, system modeling, requirements validation, requirements management, and requirements engineering for critical systems. These guidelines are further divided into three categories: 36 basic guidelines, 21 intermediate guidelines, and 9 advanced guidelines. The assessment computes a weighted score of how well guidelines are adopted within the organization. Assessors determine whether each guideline is not used, discretionary based upon the project manager, normally used, or standardized throughout the organization. To achieve the highest possible score, all guidelines should be standardized throughout the organization.

The weighted score for basic, intermediate, and advanced guidelines is used to compute an organization's RE maturity level: initial (Level 1), repeatable (Level 2), or defined (Level 3). The scoring mechanism for REGPG does not dictate that an organization has to pass all guidelines to reach a certain maturity level; instead, it offers some flexibility for organizations to pick guidelines that are meaningful for their specific context. Level 1 organizations have limited use and standardization of basic guidelines. As with the SW-CMM, Level 1 organizations rely on the expertise of people within the organization instead of standardized processes. As such, they may be unable to consistently deliver quality results and requirements issues are common. Level 2 organizations have a higher weighted score for basic guidelines and some proficiency with intermediate and advanced guidelines. This higher score indicates that more guidelines are being followed within the organization and requirements problems are likely to be reduced. Finally, Level 3 organizations master most of the basic guidelines and have higher levels of proficiency

[1] The idea of combining prescription and perspective-based methods and very early experiences from *TelSoft* were presented at the America's Conference on Information Systems (AMCIS 2006); this paper, however, is the first complete presentation of the Combined RE Process Assessment Method and the pre- and post-assessment results at *TelSoft*.

in intermediate and advanced guidelines. They have consistently defined and standardized RE processes in several areas throughout the organization.

The REGPG has been used to assess ERP RE processes [32], [33], to develop a formal assessment instrument [34], and to suggest general success criteria for RE improvements [35], [36]. Sommerville and Ransom [37] provide recommendations for adapting the method such as having domain-specific assignment of processes to maturity levels, creating domain-specific versions of the underlying model, and focusing on the business benefits of improving RE processes.

While there are important variations across these assessment methods, they all analyze the gap between generic RE process prescriptions and current process practices. A process model drives data collection and analysis, specifies which processes should be adopted, and outlines priorities to effectively increase RE maturity. Although all three methods acknowledge the importance of tailoring assessments to organizational needs, they are prescriptive in nature and assume that RE is assessed and improved by benchmarking against the best practices [5], [6].

2.2 The Role of Stakeholder Perceptions

Stakeholder perceptions and subjective judgments have historically played important roles in various software disciplines: Host and Wohlin [13] provide insights from a subjective effort estimation experiment; Hughes [14] presents expert judgment as an estimation method; Ropponen and Lyytinen [16] identify the main components of software development risk based on the perceptions of experienced project managers; Lyytinen et al. [38] discuss software risk management methods as attention shaping mechanisms that impact the perception of key stakeholders; stakeholder perceptions play a major role in postmortem evaluations of software projects [39], [40]; and Lewis et al. have developed an approach to business process innovation based on stakeholder perceptions [15]. These applications of stakeholder perceptions are typically complemented with objective data and offer specific advantages.

Leveraging stakeholder perceptions in RE process assessment can enhance the specificity of the knowledge gained and the recommended actions. First, stakeholder perceptions can provide important knowledge about traditions and values that shape practices within a particular company. Second, expert perceptions of key problems can help prioritize recommended actions within the company. Finally, including stakeholders' perceptions of problems into assessments can increase commitment to subsequent changes.

For these reasons, researchers have engaged in developing inductive assessment methods based on stakeholder perceptions of current practices [5], [7]. Nielsen and Nørbjerg [17] report research in which stakeholders in low-maturity companies were actively engaged in identifying key problems and implementing what was perceived to be sensible improvements; Wohlin and Andrews [12] present a method for analyzing project success based on subjective data; Liu et al. [11] provide a method to prioritize process assessments from multiple stakeholders and aggregate them into requirements for improvement; Frederiksen and Mathiassen [10] present an information-centric method based on stakeholder participation for assessing software metrics programs; and Pettersson et al. [7] have developed a lightweight method for software process assessment and improvement planning based on involvement of key stakeholders. While these methods document the feasibility of basing assessments on stakeholder perceptions, none of them focuses specifically on RE, and none of them provides guidance on how to effectively combine perceptions with prescriptions.

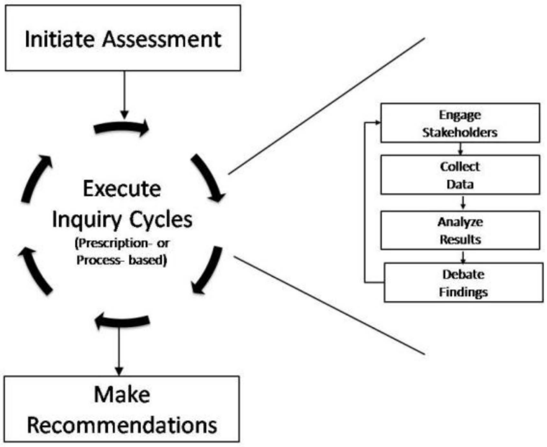

Fig. 1. Combined RE process assessment method.

Therefore, this research was designed to contribute to the RE assessment literature and the general software process assessment literature by addressing the following research question:

- How can stakeholder perceptions and process prescriptions be combined during assessment to effectively balance specificity, comparability, and accuracy challenges in RE process improvement?

3 A Combined Method

A combined method for RE process assessment has the potential of drawing upon two distinct sets of advantages. By relying on prescriptive processes, a combined method can make comparisons across companies feasible; can support a structured and easy-to-adopt assessment approach; and can lead to an immediate set of recommendations for improvement. By relying on stakeholder perceptions of current process practices, a combined method can help understand the specific characteristics and priorities within the company, can tailor recommendations to the specific company context, and can engage stakeholders in ways that increase buy-in and facilitate successful implementation of new processes. In summary, this is the rationale for combining perceptions and prescriptions in software process assessments.

Our combined methodology consists of three steps: initiating the assessment, executing multiple inquiry cycles, and making recommendations based upon findings. The method is illustrated in Fig. 1 and its key activities and related research are summarized in Table 1.

First, the RE process assessment is initiated. The objective is to establish commitment, select an assessment strategy, and agree on an overall plan for the inquiry cycles

TABLE 1
Key Activities and Related Research

Step	Activities	Research
Initiate Assessment	Establish commitment Select assessment strategy Develop assessment plan	[5], [6], [7], [19], [36], [41]
Execute Inquiry Cycles	Engage stakeholders Collect data Analyze data Debate findings	[7], [10], [11], [12], [17], [23], [24], [25], [26]
Make Recommendations	Appreciate context Prioritize recommendations Develop improvement plan	[4], [19], [36], [37], [41]

and the recommendation step, see Table 1. Prior literature has identified several success factors for RE process improvement, including management support, motivation and commitment of other employees, and a systematic implementation strategy [36]. As a first activity, it is therefore important to ensure management commitment and sponsorship, to actively involve key stakeholders, and to have management agree on the overall plan that results from this first step [19], [41].

Selection of assessment strategy should consider how to prioritize assessment comparability and assessment specificity [5], [7] by asking the questions: How important is it that results are complete and comparable to the best practices? How important is it that the specific characteristics of this company's processes and needs are taken into account? Weighing these concerns against each other can be done by assigning an emphasis in percentage to each, with 50-50 percent indicating a strategy in which equal emphasis is put on the two concerns.

Developing an overall assessment plan should be based on the chosen strategy and consideration of assessment accuracy [6]. Desired accuracy can be a light weighting (days of effort relying on a few key sources), a medium weighting (weeks of effort relying on many different sources), or a heavy weighting (months of effort relying on complete coverage of issues and stakeholders). In developing the assessment plan, the chosen strategy will help decide on a set of inquiry cycles, where some cycles are based on stakeholder perceptions and some on process prescriptions; the strategy will help sequence cycles into a useful overall plan in which sources and data from one cycle can be considered as basis for subsequent cycles, and the strategy will help specify which resources, stakeholders, and other sources to involve in each cycle. The key output of the first step is commitment from management and other key stakeholders to an overall assessment plan.

Second, current RE process practices are assessed through the series of inquiry cycles specified in the assessment plan. Each inquiry cycle, whether perception-driven or prescription-driven, involves engaging stakeholders, collecting data, analyzing data, and debating findings, see Table 1. Planning each cycle in detail involves selection of stakeholders to engage, selection of projects to analyze, and selection of sources of data (documents, observation, interviews, and workshops) [7], [10].

Perception-driven inquiry captures data about individual beliefs and experiences in the specific company context; the detailed planning of these cycles can find inspiration in a number of inductive assessment methods: systematically selecting projects and people and triangulating between interview data and project documentation [7], engaging stakeholders in structured debates of different system models of current practices [10], systematically prioritizing perceptions across stakeholder groups [11], using survey data to rank projects according to specific success variables [12], and identifying and prioritizing perceived problems based on interviews with key project managers [17], [7], [10], [11], [12]. Prescription-driven inquiry captures data on how current practices benchmark against predefined processes, best practice, and predefined questions; there are a number of RE-focused prescriptive methods available for this purpose: analyzing RE related data based on SW-CMM mini assessments [24], using the RE version of SW-CMM, R-CMM [23], [24], [25], [26]. The outcomes from the second step include a list of problems as well as opportunities for improvement.

Third, the knowledge gained from the inquiry cycles is used to make recommendations. The challenge is to turn insights from the inquiry cycles into recommendations for improvement that are well aligned with the software company's priorities, traditions, and culture [4], [41]. Also, while it can be challenging to provide hard evidence of improvements, it is important to effectively communicate to colleagues the likely benefits of the proposed improvement plan [36], [37]. Making recommendations requires three activities: appreciating the context for improvement, prioritizing recommendations, and developing a plan for improvement, see Table 1.

The outcome from the third step (Make Recommendations) is a commitment from management and other key stakeholders to an overall RE process improvement plan. The plan should identify the process areas to be improved; suggest an overall strategy for managing the change process, establish project teams that focus on improving each identified process area, and consider the appropriate sequencing of the improvement efforts [19].

4 Industrial Case Study

The research we present is part of an action research [42] project carried out between *TelSoft* and researchers from the Center for Process Innovation (CEPRIN) at Georgia State University. This industrial case allowed us to investigate in a real-world context [43] how to combine perceptions and prescriptions in RE assessment. It also allowed us to evaluate the impact such an approach can have on RE maturity and perceived business benefit. In this section, we provide background information about the research site and approach.

4.1 Research Site

TelSoft was founded in 1971, with the mission to be a premier software services firm in the telecommunications and utility industries. The company has approximately 500 employees with 50 dedicated to software development. Many of the same employees who helped found the organization 37 years ago are still employed, bringing both a wealth of experience and old habits. One of the authors had previously worked at *TelSoft*, which allowed the research team immediate and deep engagement. It also provided a solid understanding of the context and acceptance of the research collaboration by *TelSoft* employees.

TelSoft emerged as an ideal site because the company was experiencing significant problems related to RE issues. For example, *TelSoft* depended on a few very large customers that constantly required software engineers to respond to requirements changes. Also, these customers had different requirements elicitation and documentation processes in place and *TelSoft* was expected to adapt to each of these. Finally, software releases were often shipped with deviations from agreed upon requirements. *TelSoft* had previously been engaged in improving RE practices through a CMM-based initiative. While this effort resulted in documented new processes, these processes were not appropriate for the culture and business realities at *TelSoft*. Therefore, no sustainable changes had been implemented.

4.2 Research Approach

The study was designed as collaborative action research between *TelSoft* and the authors [27]. The goal of action research is to "contribute both to the practical concerns of people in an immediate problematic situation and to the goals of social science by joint collaboration" [44]. Action research approaches differ according to their process model (iterative, reflective, or linear), structure (rigorous or fluid), typical involvement (collaborative, facilitative, or experimental), and primary goals (organizational development, system design, scientific knowledge, or training) [45]. Action research typically follows a learning cycle that consists of diagnosing, action planning, action taking, evaluating, and specifying learning [42]. Action research leads to rich data collected through techniques such as tape recordings of meetings, participant observation, interviews, document analysis, and surveys; the resulting data provide a strong foundation for supporting research that is high in external validity and relevance. Such characteristics make action research an excellent candidate for studying longitudinal organizational change processes [46].

For this project, we selected a specific form of action research called collaborative practice research (CPR) [27]. CPR aims to understand and improve software practices through close collaboration between academic researchers and practicing engineers and managers. The collaboration was initiated in August 2004; it followed the IDEAL model for improving software processes [41] (see Fig. 2) and it was terminated in March 2007.

The Initiating phase ended in November 2004; the Diagnosing phase ended in May 2005 and included the first assessment of RE processes reported in this paper; the Establishing and Action phases ended in December 2006 and included a number of dedicated projects to improve RE processes based on the recommendations from the assessment. Finally, the Learning phase ended in March 2007 and involved the second assessment of RE processes included in this paper.

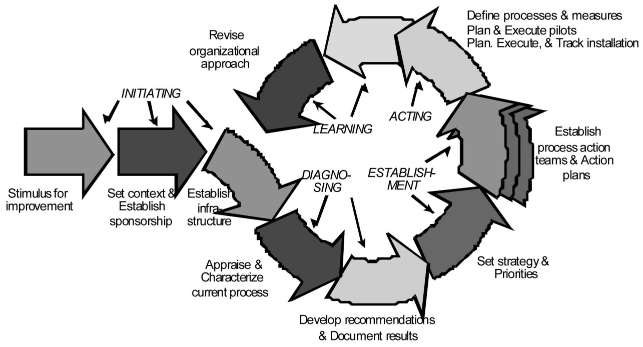
Revise organizational approach
Define processes & measures
Plan & Execute pilots
Plan. Execute, & Track installation
INITIATING
LEARNING
ACTING
Establish process action teams & Action plans
Stimulus for improvement
Set context & Establish sponsorship
Establish infra-structure
DIAGNO-SING
ESTABLISH-MENT
Appraise & Characterize current process
Set strategy & Priorities
Develop recommendations & Document results

Fig. 2. The research process [41].

The research-industry collaboration was organized into three teams: a Steering Committee (SC), a Problem Solving Team (PST), and a research team. The SC was composed of three researchers and *TelSoft*'s chief executive officer, division president, and vice president of software development. Meetings were held on a quarterly basis and used to set strategic direction for improvement of RE processes. The PST was composed of three researchers, *TelSoft*'s vice president of software development, and three mid-level managers. The PST met monthly to manage operational aspects of the improvement initiative. The research team consisted of the three authors from CEPRIN who typically met biweekly to discuss interim research findings and manage the collaboration.

To ensure the validity of action research, it is important to exhibit rigor during data collection and analysis. Here, action researchers can learn from general recommendations for qualitative research such as techniques for documenting field notes [47], facilitating data analysis through computer software [48], and demonstrating traceability between data and results [49], [50]. Developing and maintaining a case study database enhance the reliability of qualitative research and make an independent audit of claims possible [28], [47]. In the *TelSoft* case, data were systematically collected throughout the project, both to support the improvement effort at *TelSoft* and to support the evaluation of the combined RE process assessment method. Interviews, workshops, and meetings were recorded whenever feasible for later transcription. A designated note taker created public meeting minutes for most PST and improvement team meetings. Other data sources included reflective field notes from research team members, process documentation from *TelSoft*, and e-mail messages between CEPRIN and *TelSoft*.

In addition to systematic data collection, the research team took other measures to ensure the rigor of data analysis. First, our conclusions were drawn from a variety of data sources, which helped enhance the reliability of results [28], [47], [51]. Second, the research team composition balanced the insider perspective of the former *TelSoft* employee with the outsider perspective offered by two researchers new to *TelSoft*. Third, the research team created interim reports and validated findings with key stakeholders from *TelSoft*. Finally, the research team met to review summary documents, abstract major themes, and document evidence that supported each theme. A complete account of the research project and evaluation can be found in [52].

5 Pre-Assessment at *TelSoft*

The RE assessment method was used twice at *TelSoft*: once at the beginning to Diagnose [41] current problems with RE processes and again at the end of the collaboration to Learn [41] about the impact of the RE improvement. In this section, we describe the pre-assessment at *TelSoft*. More specifically, we detail how we collected, analyzed, and interpreted data on RE processes at *TelSoft*, how we arrived at recommendations for improvement, and how these recommendations were implemented though a number of process action teams.

5.1 Initiate Assessment

From November 2004 to May 2005, we conducted an RE assessment at *TelSoft* using the combined method. The effort involved 22 semi-structured interviews, two 3-hour workshops, a standardized assessment, and nearly a dozen meetings of the PST. The RE assessment was managed by the PST.

The goals were to determine strengths and weaknesses of the existing RE process practices and identify improvement opportunities. Based upon these goals and *TelSoft's* previous experience with 100 percent prescription-driven assessments and improvements, it was decided to follow a strategy in which assessment specificity and comparability were emphasized 75-25 percent. *TelSoft* was strongly committed to the effort and decided to give assessment accuracy a heavy weighting (i.e., months of effort relying on complete coverage of issues and stakeholders). The PST identified three stakeholder groups involved in creating and managing requirements: software development, internal customers, and external customers. The PST decided to plan one inquiry cycle for each of these groups. To compare the company's RE processes against best practice, a prescription-driven inquiry cycle was also included in the assessment plan and the PST decided to adopt the REGPG [26] for that purpose.

5.2 Execute Inquiry Cycles

The resulting assessment plan contained three perception-driven inquiry cycles and one prescription-driven cycle. Key insights from each of these cycles are summarized in the following.

5.2.1 Inquiry Cycle 1: Software Development Perceptions

The Software Development group at *TelSoft* is responsible for interacting with clients to generate specifications of software requirements, creating software based upon these requirements, evaluating the impact of requirements changes, and ensuring the quality of the product. An interview guide was created that asked structured and semi-structured questions about both objective and subjective requirements-related documentation and activities. Sample questions included the following:

1. Requirements documents: Which requirements documents do you interact with? From whom do you receive information about requirements? What form do those requirements take? How would you assess the quality of these documents?
2. Requirements activities: How are you involved in documenting or managing requirements? To what extent do you collaborate with others to understand requirements? To whom do you send information about requirements?
3. Open-ended: What strengths do you see in the way requirements are documented or managed at *TelSoft*? What weaknesses and opportunities for improvement do you see?

We interviewed nine representatives from the Software Development group: two project managers, two software engineers, one quality assurance analyst, two business analysts, and two mid-level managers. The interviews typically lasted one hour and were attended by the first author and another member of the research team. After conducting a series of interviews, the research team met to reflect upon emerging themes related to RE practices at *TelSoft*. A summary of each interview, along with these observations, was added to the case study database for later analysis. After all interviews had been conducted, the research team again met to analyze data. To support identification of possible improvements, our analysis focused on the weaknesses identified. Participants' perceptions were analyzed to identify similar themes and documented in a list of 17 potential problem areas. Later, all members of the Software Development group participated in a three-hour workshop to evaluate and validate this list. For each problem area, participants individually provided an assessment of criticality, feasibility, and priority. The individual responses were then debated and again prioritized in breakout sessions during the workshop. A plenary session was finally held in which all groups described their top issues. Table 2 presents the RE problems that the Software Development group gave their highest averaged rankings.

5.2.2 Inquiry Cycle 2: Internal Customer Perceptions

In the second perception-driven cycle, we focused on the internal groups that interacted with the Software Development group to generate and manage software requirements. The Software Development group receives requirements from both the marketing organization and an internal production group that uses its software. We interviewed two sales people, three project managers for the internal production group, and a mid-level manager. Once the interviews were completed, the authors again analyzed the interview data for common themes that suggested potential problem areas. We held a workshop for validating and prioritizing the 14 identified problem areas that involved the people interviewed as well as other internal customers. Table 3 lists the RE problems given highest priority by internal customers.

5.2.3 Inquiry Cycle 3: REGPG Assessment

Through the first two inquiry cycles, we learned of key concerns related to requirements process practices from the perspective of *TelSoft* employees. However, we also wanted to evaluate *TelSoft* against best practices to uncover additional vulnerabilities and opportunities. The REGPG assessment [26] as described in Section 2.1 was chosen

TABLE 2
Software Development Problem Areas

Problem Area	Description
QA Disintegration	Quality assurance department needs to be kept informed as detailed requirements evolve.
Change Management	Requirements changes are not addressed in a systematic fashion; documents are not kept updated and consistent.
Ad-Hoc Review	Review of requirements is often performed in an ad-hoc fashion where reviewers are unprepared and critique is not systematically fed back into the requirements process.
Resource Allocation	Quality assurance and core development have difficulties in prioritizing tasks and requests across projects.
Customer Variation	There are significant variations in requirements management and quality assurance practices across customers
Process vs. Practice	*TelSoft*'s documented requirements management process is considerable different from actual practice; the ongoing maintenance and innovation of the described processes is not institutionalized.
Documentation Standards	Documentation standards vary; there are considerable variations in style and level of detail across authors; the most appropriate documentation form is not necessarily chosen to effectively target documentation users; some documentation standards do not fit current needs.
Outdated Tools	Tools and methodologies for requirements management are not state-of-the-art; there are no procedures or responsibilities in place to facilitate improvements.

TABLE 3
Internal Customer Problem Areas

Problem Area	Description
Unsystematic early capture of requirements	*TelSoft*'s sales and marketing representatives often capture client requirements in unsystematic, non-documented ways as a basis for later interaction with other *TelSoft* stakeholders.
Changes not systematically communicated to internal users	Procedural and software changes are not systematically communicated to internal users.
Difficulty pricing based on requirements documentation	One important intention with requirements documentation is to support pricing during the bid process. However, most clients spend little time specifying requirements upfront. As a result, they often provide incomplete requirements leading to inaccurate pricing.
Complex chain of requirements communication	There are several *TelSoft* stakeholders (e.g., sales, project management, business analysts, and software developers) involved in the requirements process. That leads to many interpretations and necessary translations, each introducing new sources of error.

because prior empirical research showed it to be useful for RE process improvement (e.g., [53]). Additionally, the authors had access to an REGPG assessment tool [37] that simplified data collection, provided process guidance, ensured accurate calculation of requirements maturity, and automated report generation.

The assessment was conducted during a meeting with members of the PST who all shared their insights gained during the two first-assessment cycles. An assessment summary is shown in Table 4. Participants were provided in advance with a written description of each of the 66 practices and expected benefits of including the assessment practice. Early on, the group eliminated practices associated with the critical systems area as unnecessary for *TelSoft*'s business. Each relevant practice was read aloud and categorized by consensus of the PST members as being standardized, normalized, discretionary, or never followed. During discussion, the group created an additional category called "standardized but not checked" to indicate that *TelSoft*'s documented processes met the spirit of the practice, but there was no mechanism in place to ensure compliance. For the purpose of calculating RE maturity, this was coded as standardized in the REGPG assessment tool. For questions the group did not feel prepared to answer based on their current knowledge, PST members solicited responses from appropriate individuals after the meeting. When all practices had been evaluated, we assessed the usefulness of this assessment—what we learned, what possible actions could be taken, and how this compared to what we had discovered from the two workshops.

The company's overall RE maturity level was assessed at Level 1, Initial. In fact, *TelSoft* used only 19 of the 36 possible basic guidelines, 9 of the 21 intermediate guidelines, and none of the advanced guidelines. Since the usage of basic and intermediate guidelines was often discretionary, the weighted score fell short of the amount needed for *TelSoft* to be considered a Level 2 organization. Hence, *TelSoft* could improve its RE maturity level by consistently applying guidelines already in use and by adopting new basic guidelines throughout the organization.

TABLE 4
REGPG Assessment Summary (30/3/2005)

	Basic	**Intermediate**	**Advanced**
Guidelines Used	19	9	0
Weighted Score	37	14	0
Max Possible	105	66	27
Score % of Max	**35%**	21%	0%
Level	**Initial**		

TABLE 5
External Customer Problem Areas

Problem Area	Description
Configuration management	*TelSoft* needs to increase the transparency and consistency of its configuration management, documentation, and test activities.
Software packaging	*TelSoft* needs to improve its packaging procedures and related release notes.
Communication with client	*TelSoft* needs to increase the frequency and consistency of their communication with the client.
Project estimates	*TelSoft* needs to make better early estimates to help scope projects.

TABLE 6
Software Policies at TelSoft

ID	Description	Related Problem Areas
1	*TelSoft* will strive to operate based on the highest professional standards and processes	QA disintegration, Ad-hoc review, Resource allocation, Process management, Complex chain of requirements communication
2	*TelSoft* will strive to understand and incorporate its customers' business knowledge in our products.	Unsystematic early capture of requirements
3	*TelSoft* will maintain a proactive professional relationship to its customers.	Customer variation, Software packaging
4	*TelSoft* will manage each development project with a two-phase approach that separates requirement and development activities	Project estimates
5	*TelSoft* will only engage resources to start design and construction when TelSoft has a baseline of identifiable and agreed upon requirements.	Documentation standards
6	*TelSoft* will only engage resources to address requirement change requests that are documented, agreed upon and applied to the requirements baseline	Documentation standards, Change management, Changes not systematically communicated to internal users
7	*TelSoft* will communicate status to its customers of all active projects on a regular basis.	Communication with client
8	*TelSoft* will only deliver official releases of software to a client with the written approval of Quality Assurance	Configuration management
9	Each release of *TelSoft* software will include documentation of all changes and new features since the previous release.	Configuration management

The REGPG assessment also evaluated *TelSoft*'s performance in the eight RE key areas. This evaluation indicated that *TelSoft* could improve in analyzing, validating, and managing requirements. This finding echoed some of the issues described in earlier inquiry cycles, particularly as it relates to managing requirements changes. The REGPG assessment identified *TelSoft*'s strengths as being in the areas of documenting, eliciting, and describing requirements. This assessment stood in direct contrast to findings from earlier inquiry cycles. For example, stakeholders in the Software Development group felt that the form and level of detail for requirements documentation was not always appropriate. Some internal customers complained of unsystematic early capture of requirements that lead to inaccurate cost estimates and misperceptions of project scope.

5.2.4 Inquiry Cycle 4: External Customer Perceptions

In the final inquiry cycle, we interviewed external customers who interacted with *TelSoft* to generate software requirements, request requirements changes, and perform user acceptance testing. The PST selected seven client representatives from three of *TelSoft*'s longtime customers. A new interview guide was created that asked about requirements documentation, requirements management, and process innovation. In this cycle, there was no workshop used as a discussion forum. The customers praised *TelSoft* personnel for understanding their business, responding promptly to customer requests, and adapting internal practices to client's needs. Table 5 summarizes the areas for improvement as identified by external customers.

5.3 Make Recommendations

The combined assessment revealed that *TelSoft* needed to develop its ability to sense customer needs and respond to technological and market opportunities. *TelSoft* also needed to be more proactive in interactions with customers, sharing information about software development to increase client confidence in the software product. Based upon this assessment, it was recommended that *TelSoft* combine documented processes to support a more disciplined and structured approach to RE [54], [55] with management principles to strengthen their ability to adapt to changing customer needs and emerging market opportunities [45], [48], [49], [50].

One of the first actions taken was the creation of core policies for software development, see Table 6. The PST worked collaboratively with *TelSoft* employees at all levels of the organization to develop and establish these policies. While some policies are directly related to RE (i.e., 2, 4, 5, 6, and 9), others address broader software development issues (i.e., 1, 3, 7, and 8). The policies served as general operating principles and guidance for further improvement efforts. For example, consistent noncompliance with one of the company policies would indicate that additional resources should be given to make improvements. Table 6 shows how the policies address the identified problem areas. The next section provides examples of how the policies helped shape the intervention at *TelSoft*.

5.4 Resulting Actions

The improvement strategy was implemented through focused and dedicated project teams with clear success criteria and specified deliverables. These project teams were established, monitored, and coordinated through the PST. Once the SC approved the proposed project teams, a kickoff seminar presented the assessment results to all employees in the Software Development group to validate findings and create additional input from employees on suitable improvement activities. The PST created two separate intervention cycles of Establishing and Acting [41], which lasted from 8 to 10 months each.

During the first intervention cycle, five improvement teams were formed to implement policies and eliminate problem areas: software coordination, quality assurance, configuration management, customer relations, and requirements management:

- *Software coordination.* To facilitate systematic, professional oversight (Policy 1), the software coordination team committed to monthly review of organizational strategy, software projects, improvement activities, and emerging software requirements and innovation opportunities.
- *Quality assurance.* To support more efficient operation (Policy 1), the quality assurance team redesigned standard operating procedures for certification, regression, and acceptance testing based on agreed upon requirements.
- *Configuration management.* To support the need to fully document each release of *TelSoft* software (Policy 9), the configuration management team developed a software release specification template to document the source code, software requirements, build procedures, and configuration files associated with each release.
- *Customer relations.* To move toward a more proactive communication style (Policies 3 and 7), the customer relations team communicated information about the improvement activities with customers and began to address customer concerns identified during Inquiry Cycles 2 and 4.
- *Requirements management.* Given the increased importance of having agreed upon and documented requirements before proceeding with design and construction (Policies 5 and 6), the requirements management team simplified the requirements specification to reduce the number of required sections and created a change control template to be used for all changes to requirements.

During the second intervention cycle, the improvement teams focused on customer relations, quality results, and process management as follows:

- *Customer relations.* To help maintain a proactive professional relationship with customers (Policy 3), the customer relations team purchased contact management software for sales representatives and management, redesigned the packaging for software releases, and developed guidelines for engaging customers from the proposal through the deployment stages.
- *Quality results.* To increase the likelihood that customers would receive a quality product (Policy 8), the quality results team developed guidelines for conducting postproject analysis to determine root cause of problems, cleaned up the software defect database, and improved the efficiency of the regression testing.
- *Process management.* To help consistently operate based upon professional standards and processes (Policy 1), the process management team created standards for process documentation, evaluated existing process descriptions in relation to these standards and future use at *TelSoft*, as well as developed a plan for process management.

6 Post-Assessment at *TelSoft*

In this section, we describe the assessment that occurred at the end of the industry-research collaboration to Learn [41] about the impact of the RE improvement efforts. We detail the collection, analysis, and interpretation of data on the new RE processes at *TelSoft*.

6.1 Initiate Assessment

From December 2006 to March 2007, we conducted a post-assessment using the combined method. Data collection methods included interviews, an online questionnaire, and a standardized RE assessment tool. This assessment evaluated SPI impact (the goal was to identify changes in each of the seven improvement areas, the effect of the software policies on day-to-day practice, challenges that occurred in enacting changes, and suggestions for improvement), SPI organization (the goal was to assess how effectively the PST, SCG, and improvement teams had managed the SPI effort), and SPI perceptions (the goal was to determine how different stakeholders perceived the overall value of the SPI effort, their satisfaction with their own level of involvement, and the usefulness of communication methods used).

Based on these goals, the assessment strategy would again emphasize assessment specificity over comparability. Assessment accuracy was given a medium weighting (i.e., weeks of effort). We identified four major stakeholder groups: customers, improvement team participants, SPI leadership (SC and PST), and other software development employees. To be able to compare the company's progress in RE maturity, the REGPG standardized assessment was repeated.

6.2 Execute Inquiry Cycles

The resulting assessment plan consisted of two perception-driven (interviews and questionnaire) and one prescription-driven (REGPG assessment) inquiry cycle. Next, we briefly summarize key findings from each cycle while Section 7 compares results across inquiry cycles and reflects upon the RE assessment process.

6.2.1 Inquiry Cycle 1: Participant Perceptions

The first perception-driven inquiry cycle was based upon 10 semi-structured interviews. An interview guide was created based upon the objectives of evaluating SPI impact, SPI organization, and SPI perceptions. Sample questions included the following:

1. SPI impact: What has been the overall impact of the improvement initiative? Can you provide specific examples of how the initiative has positively impacted business?
2. SPI organization, the following questions were asked for the PST, SCG, and improvement teams: What do you see as the underlying reason for having this team? What is the main impact of this team? How effective has this team been in managing its effort?
3. SPI perceptions: How do different stakeholders perceive the SPI initiative (e.g., cynicism, enthusiasm,

indifference)? To what extent are those outside the SPI initiative informed about it? Do they need more or less information?

Three representatives from two external customers consented to phone interviews. Since a questionnaire would be sent to all employees, the PST selected only seven employees for face-to-face interviews: five managers involved in the PST and SCG plus two developers who had actively participated on improvement teams. Each interview lasted roughly 45 minutes, was audibly recorded, and later transcribed. The findings were compiled into multiple reports and shared at various levels throughout the organization. The summary of external customer interviews was provided to the PST as well as the primary customer liaison at *TelSoft*. Interview comments were combined with data from the questionnaire (described next) as part of an overall SPI impact report. This consideration of stakeholder perception provided a more complete picture on the extent to which *TelSoft* was successfully implementing its policies and improving its RE processes.

To improve its management of customer relations, *TelSoft* had increased its face-to-face time spent with customers and proactively communicated the software policies. *TelSoft*'s management had renewed their emphasis on customer relations as indicated by the Division President: "I think we've done several very specific things—realizing the importance of having a professional image, emphasizing the importance of having continuous customer interaction, and identifying opportunities throughout our work cycle to encourage and almost require more customer interaction." The customer representatives indicated that both the level of communication and quality of the resulting software product had improved.

Changes to standardize the software release process and systematically document the process for building software releases (Policy 9) were also well received by several groups. Responsibility for this process was successfully moved to the Quality Assurance group: "QA doing builds means they can trust the integrity of the builds." The QA manager further described how the detailed documentation provided early detection of a missing file from a release, saving *TelSoft* from shipping a defective product.

Although *TelSoft*'s policies required documented requirements before proceeding with new construction or changes (Policies 5 and 6), several stakeholders were still dissatisfied with the quality of the requirements documentation. Participants in the Software Development group expressed concern over the new streamlined requirements specifications. *TelSoft* also continued to act in a less disciplined manner for internal projects. While developers wanted requirements documents to provide an unambiguous view of a system, they described recent documents as having "lots of holes" with requirements being implied instead of explicitly stated.

6.2.2 Inquiry Cycle 2: Software Development Questionnaire

The second inquiry cycle was based on an online questionnaire sent to 25 *TelSoft* employees who either reported to the VP of the Software Development or had otherwise been involved in the SPI effort. The content of the questionnaire was first created by the research team and then refined and piloted by the PST. The questionnaire asked each individual to assess the impact of the overall initiative, the software policies, and the modified processes and templates. In addition, several open-ended questions allowed respondents to provide additional detail. Data from the questionnaire played a key role in the overall SPI impact report. Fig. 3 shows the majority of participants perceived the initiative as either having "some improvement" or "considerable improvement" in software practice.

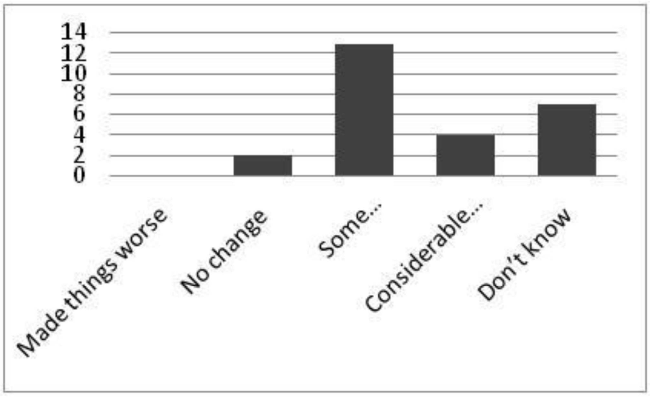

Fig. 3. Participant perception of overall impact.

Most employees realized that process improvement was a legitimate activity that received significant management support as indicated by these remarks:

- "I think people are more in tune to the fact that process is important."
- "People think more critically about our processes now as a result of attention to these issues."

Those who were more uncertain about the initiative's impact needed more time to implement changes and better communication of changes to people throughout the organization:

- "Implementation is slow, and following procedures are somewhat sporadic at times as we phase into some of the initiatives."
- "I'm not convinced all initiatives have been fully implemented."

6.2.3 Inquiry Cycle 3: REGPG Assessment

The third inquiry cycle was again based on REGPG [26]. The assessment was completed by the VP of the Software Development and the QA manager in June 2007. The values were compared against the REGPG assessment conducted on March 2005. Overall, *TelSoft*'s requirements maturity increased from Initial (see Table 4) to Repeatable (see Table 7). *TelSoft* now used 27 of the 36 basic guidelines, an increase of eight additional guidelines over the 2005 assessment. Although the number of intermediate guidelines remained the same, the weighted score was higher, indicating that *TelSoft* was more consistently applying these guidelines throughout the organization. Finally, *TelSoft* had also begun using one of the advanced RE guidelines.

TelSoft increased the percentage of the best practices used in five of eight areas, see Table 8. *TelSoft* improved all of its Weak areas to Average. As in the initial assessment, the strong REGPG assessment in the requirements document

TABLE 7
REGPG Assessment Summary (6/19/2007)

	Basic	Intermediate	Advanced
Guidelines Used	27	9	1
Weighted Score	62	19	1
Max Possible	105	66	27
Score % of Max	**59%**	29%	4%
Level	**Repeatable**		

area contrasted with the stakeholder perceptions from Inquiry Cycles 1 and 2.

6.3 Make Recommendations

Overall, *TelSoft*'s management team was pleased with the SPI initiative, as expressed by *TelSoft*'s Vice President of Software: "We are seeing the benefits of the collaboration in better portfolio planning and coordination, improved customer relations, less internal strife over requirements management, fewer quality assurance (QA) cycles and increased transparency of our configuration management." To sustain these improvements, two major recommendations were adopted.

The first recommendation was to continue having the PST meet to monitor progress and facilitate implementation of improvement efforts. After meeting over the three-year period, this group had learned to assess processes and manage action teams. Their initial focus would be on ensuring that plans from the existing action teams were effectively implemented.

The second recommendation was to focus on improving project management capability at *TelSoft*. Although *TelSoft* had become more disciplined on external projects, many internal projects continued to exceed budget and schedule. Several suggestions from employees pointed toward improving project management disciplines such as cost estimation, quality assurance, and scope management. Employees noted that processes and expectations were inconsistent across projects and project managers needed to be held more accountable. As a consequence, *TelSoft*'s management team agreed to collaborate with the research team for another year targeted at improving software project management processes.

7 Discussion

The software engineering literature categorizes methods for software process assessment and improvement into prescription-driven and perception-driven approaches [5], [7]. Prescription-driven approaches (e.g., [8], [9], [26], [30]) make comparisons across companies feasible, they support a structured and easy-to-adopt assessment approach, and they lead to an immediate set of recommendations for improvement. Perception-driven approaches (e.g., [5], [7], [10], [11], [12]) help understand the specific characteristics and priorities within the company, tailor recommendations to the specific company context, and engage stakeholders in ways that increase buy-in and facilitate successful implementation of new processes. Hence, acknowledging that there are no combined software process assessments methods available in the literature and RE is one of the key activities in SE, the objective of this research was to develop a combined method to help managers effectively balance assessment specificity, comparability, and accuracy in RE process improvement.

TABLE 8
Strength Matrix

Area	Weak	Average	Good	Strong
Requirements Document				Pre Post
Requirements Elicitation			Pre	Post
Requirements Analysis/Negotiation	Pre	Post		
Describing Requirements				Pre Post
System Modeling		Pre	Post	
Requirements Validation	Pre	Post		
Requirements Management	Pre	Post		
RE for Critical Systems	Pre Post			

As a result, we have presented a generic, combined method for assessment of RE processes (Section 4). The method consists of three steps: initiating the assessment, executing multiple inquiry cycles, and making recommendations based upon the findings, see Fig. 1. Each step consists of specific activities and draws upon the software process assessment literature, see Table 1. Based on strategic considerations of assessment specificity, comparability, and accuracy, managers can tailor the generic method to specific contexts. Moreover, managers can combine prescription-driven and perception-driven approaches by executing distinct inquiry cycles.

The results of applying the combined method to RE assessment at *TelSoft* are documented through an industrial case study [28], [47], [51]. The combined method was applied twice: to support assessment of RE processes as a basis for extensive improvement efforts (Section 5) and subsequently to assess the impact of the improvement efforts on RE processes (Section 6). Enabled by the flexibility of the method, *TelSoft* gave heavy weighting to accuracy (i.e., months of effort) during the initial assessment and medium weighting (i.e., weeks of effort) during the post-assessment. The flexibility of the method also allowed *TelSoft* to adopt and combine different assessment techniques during the two assessments. The execution of both pre and post-assessment supported continuous improvement efforts at *TelSoft* and informed the research process about the impacts of using the combined method in a real-world setting.

Our research makes two primary contributions. The first contribution is the proposed combined RE assessment method. In relation to the RE assessment literature, existing methods are all prescriptive and the presented RE assessment method is the first to draw systematically on stakeholder perceptions. In relation to the general software process assessment literature, there are a number of inductive assessment methods available (e.g., [7], [10], [11], [12], [17]), but the presented method is the first to systematically combine stakeholder perceptions and process prescriptions to inform SPI. The second contribution is the industrial case study detailing the application of the method at *TelSoft*. The data from this case study provide a number of important insights.

Cycle interactions. Prescription-driven and perception-driven assessment cycles complemented each other in important ways. However, in each of the two applications of the combined method, this required careful management of interactions between cycles. First, during the initiation of the assessment, we identified the need for individual cycles and decided on the most appropriate way to sequence these. Second, we executed each cycle separately and with relatively little sharing of information between cycles. Third, we carefully compared and contrasted results across cycles as part of making final recommendations. The separation of concerns among cycle planning, cycle execution, and making recommendations offered a useful approach to managing individual cycles and their relationships. Finally, we acknowledge that our interpretation of results was facilitated by the one author's previous employment at *TelSoft*. Having insider status not only helped us gain entry to the company, it also helped us establish trusting relationships, prompt candid responses from participants, understand the organizational context, and ask more pointed questions. Those proceeding without this level of knowledge may benefit from increasing the time spent during the early activities of the combined RE assessment (i.e., establish commitment, engage stakeholders, and collect data).

Comprehensive RE assessments. The combined approach led to a comprehensive assessment of RE processes at *TelSoft*. One form of inquiry could provide insight into an area that the other form did not address. For example, the perception-driven inquiry during pre-assessment highlighted problems in communicating requirements changes to stakeholders *(Cycle 1, Quality Assurance Disintegration; Cycle 2, Changes not systematically communicated to internal users; Cycle 4, Increase communication with client)* that were not captured during the REGPG assessment. Also, data from one inquiry type could corroborate initial findings from the other. For example, the process-driven REGPG pre-assessment identified that *TelSoft* used only two of nine suggested practices in the requirements management area, which could lead to development rework and systems that do not meet customer's expectations [26]. Similarly, the perception-driven assessment identified weaknesses in managing requirements changes *(Cycle 1, Change Management)* and in ensuring that all stakeholders understand the current requirements and the relationships between them *(Cycle 2, Complex chain of requirements communication)*. Finally, combining the two inquiry types could lead to contradictory results. *TelSoft* earned high marks with the pre-assessment REGPG for having defined a standard document structure with an optional glossary for specialized terms and a table of contents to help readers find information; the company also routinely held requirements review sessions. However, the perception-driven assessments indicated problems related to requirements documentation. For example, even though the format was standardized, it did not meet the needs of all stakeholders in the Software Development group *(Cycle 1, Documentation Standards)*. Also, during the early requirements elicitation phases, sales and marketing representatives did not systematically document client requirements in sufficient detail for other stakeholders *(Cycle 2, Unsystematic early capture of requirements)*. Contradictory results like these triggered further analyses of the challenges in existing processes and of what directions to take to improve them.

Contextual view of RE. The combined method helped *TelSoft* view RE within the broader context of its software operation. While the prescription-driven assessment supported by REGPG focused on the 66 RE practices included in the method, the perception-driven assessments were based on semi-structured interviews, surveys with open questions, and open debates in workshops. These formats allowed the participants to reflect on the relationship between RE and the wider engineering context at *TelSoft*. Some of the recommendations and subsequent improvements, therefore, introduced changes into the engineering context that would positively impact RE processes. As a result, some of *TelSoft*'s adopted software policies (see Table 6) are directly related to RE (i.e., Policies 2, 4, 5, 6, and 9) while others address broader software development issues (i.e., Policies 1, 3, 7, and 8) to create an improved context for RE.

RE assessment impact. The data from *TelSoft* document that the combined method led to actual improvements in RE. The REGPG assessments (see Tables 4 and 7) showed that *TelSoft*'s overall requirements maturity increased from Initial (Level 1) to Repeatable (Level 2). *TelSoft* increased the percentage of the best practices used in five of the eight requirements areas and improved all of its weak areas to average (see Table 8). Although some employees were uncertain about the impact of the initiative, the majority agreed that the SPI initiative created either "some improvement" or "considerable improvement" in software practices. As a consequence, more employees perceived process improvement as a legitimate activity worthy of management support and commitment from developers.

Although these insights demonstrate the usefulness of applying the combined method at *TelSoft*, the study has limitations that point toward opportunities for future research. The first limitation relates to the adopted industrial case study based on a single software company with specific characteristics (e.g., low software process maturity, internal client challenges, and limited external market). Using a single case is appropriate given the research objective, the characteristics that *TelSoft* shares with other software organizations, and the benefits of being able to explore longitudinal data [28]. However, this approach also limits our ability to assess the usefulness of the combined method across different types of software organizations.

A second limitation relates to the facilitative role of the research team: The RE assessments at *TelSoft* were conducted in close collaboration between *TelSoft* employees and CEPRIN researchers. This approach is fundamental to action research and it allowed the researchers to get firsthand and deep insight into practice [27]. At the same time, this approach made it hard to isolate the impact of the researchers from the impact of the method itself.

A third limitation was that both assessments were tailored to the same goals and needs at *TelSoft*. While the combined approach allowed tailoring with respect to specificity, comparability, and accuracy, the pre- and post-assessment strategies at *TelSoft* both favored specificity over comparability. As a result, we had limited experiences with different options for tailoring the combined method to combinations of assessment criteria.

In response to these limitations, we have taken steps to help others adapt the method and the lessons learned to other contexts: 1) We have included a detailed description of the combined method; 2) we have provided criteria for adapting the method to other contexts; and 3) we have included a detailed description of the setting and the application of the method at *TelSoft*. As further studies are conducted using the combined method, we will be able to evaluate its applicability for a variety of settings and actors. Such studies will also allow us to develop more detailed guidelines and recommendations for selecting effective combinations of assessment techniques.

8 Conclusion

We have presented a combined method for RE assessment and illustrated its use in a case study at *TelSoft*. The method is rooted in our existing knowledge on software process assessment (e.g., [5], [7], [8], [9], [5], [7], [10], [11], [12], [26], [30], [26], [37]) and in a well-documented tradition for leveraging stakeholder perceptions and expert judgments in various software disciplines (e.g., [13], [14], [15], [16], [39], [40], [56]). The practical use of the combined method is documented through a detailed industrial case study at *TelSoft.*

Before software managers adopt the combined method, they should consider the differences in context as compared to *TelSoft* and adapt the method accordingly. The experiences from *TelSoft* do, however, suggest a few fundamental lessons on combined RE process assessment as follows:

1. Carefully initiating the assessment effort helps create solid organizational support.
2. Considering assessment specificity, comparability, and accuracy helps prioritize and plan the assessment effort.
3. Separating concerns into separate inquiry cycles helps combine prescription- and perception-driven assessment approaches.
4. Comparing and contrasting findings from different inquiry cycles help create a solid and rich basis for making recommendations.
5. Complementing the initial assessment with a post-assessment helps integrate the effort into a continuous improvement program.

Acknowledgments

The authors wish to thank the employees at *TelSoft*. This work was supported in part by a grant from *TelSoft* and the Georgia Research Alliance.

References

[1] G. Kotonya and I. Sommerville, *Requirements Engineering Processes and Techniques.* John Wiley & Sons, 1998.

[2] S. Beecham, T. Hall, C. Britton, M. Cottee, and A. Rainer, "Using an Expert Panel to Validate a Requirement Process Improvement Model," *J. Systems and Software,* vol. 76, no. 3, pp. 251-275, 2005.

[3] R. Schmidt, K. Lyytinen, M. Keil, and P. Cule, "Identifying Software Project Risks: An International Delphi Study," *J. Management Information Systems,* vol. 17, no. 4, pp. 5-36, 2001.

[4] L. Mathiassen, O.K. Ngwenyama, and I. Aaen, "Managing Change in Software Process Improvement," *IEEE Software,* vol. 22, no. 6, pp. 84-91, Nov./Dec. 2005.

[5] P.A. Nielsen and J. Pries-Heje, "A Framework for Selecting an Assessment Strategy," *Improving Software Organizations: From Principles to Practice,* L. Mathiassen, J. Pries-Heje, and O. Ngwenyama, eds., Addison-Wesley, 2002.

[6] M. Sanders and I. Richardson, "Research into Long-Term Improvements in Small- to Medium-Sized Organisations Using SPICE as a Framework for Standards," *Software Process Improvement and Practice,* vol. 12, pp. 351-359, 2007.

[7] F. Pettersson, M. Ivarsson, T. Gorschek, and P. Ohman, "A Practitioner's Guide to Light Weight Software Process Assessment and Improvement Planning," *J. Systems and Software,* vol. 81, no. 6, pp. 972-995, 2007.

[8] CMMI Product Team "CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing," CMU/SEI-2002-TR-011, Software Eng. Inst., Carnegie Mellon Univ., 2002.

[9] ISO/IEC, *ISO/IEC 15288:2002 Systems Engineering—Systems Life Cycle Processes,* 2002.

[10] H.D. Frederiksen and L. Mathiassen, "Information-Centric Assessment of Software Metrics Practices," *IEEE Trans. Eng. Management,* vol. 52, no. 3, pp. 350-362, Aug. 2005.

[11] X. Liu, Y. Sun, C.S. Veera, Y. Kyoya, and K. Noguchi, "Priority Assessment of Software Process Requirements from Multiple Perspectives," *J. Systems and Software,* vol. 79, pp. 1649-1660, 2006.

[12] C. Wohlin and A.A. Andrews, "Prioritizing and Assessing Software Project Success Factors and Project Characteristics Using Subjective Data," *Empirical Software Eng.,* vol. 8, pp. 285-308, 2003.

[13] M. Host and C. Wohlin, "A Subjective Effort Estimation Experiment," *Int'l J. Information and Software Technology,* vol. 39, no. 11, pp. 755-762, 1997.

[14] R.T. Hughes, "Expert Judgment as an Estimating Method," *Information and Software Technology,* vol. 38, pp. 67-75, 1996.

[15] M. Lewis, B. Young, L. Mathiassen, A. Rai, and R. Welke, "Business Process Innovation Based on Stakeholder Perceptions," *Information, Knowledge, Systems Management,* vol. 6, nos. 1/2, pp. 7-27, 2007.

[16] J. Ropponen and K. Lyytinen, "Components of Software Development Risk: How to Address Them? A Project Manager Survey," *IEEE Trans. Software Eng.,* vol. 26, no. 2, pp. 98-112, Feb. 2000.

[17] P.A. Nielsen and J. Nørbjerg, "Assessing Software Processes: Low Maturity or Sensible Practice," *Scandinavian J. Information Systems,* vol. 13, pp. 23-36, 2001.

[18] B. Curtis and M. Paulk, "Creating a Software Process Improvement Program," *Information and Software Technology,* vol. 35, nos. 6/7, pp. 381-386, 1993.

[19] W.S. Humphrey, *Managing the Software Process.* Addison-Wesley, 1989.

[20] T.P. Rout and A. Tuffley, "Harmonizing ISO/IEC 15504 and CMMI," *Software Process Improvement and Practice,* vol. 12, pp. 361-371, 2007.

[21] F.G. Wilkie, F. McCaffery, D. McFall, N. Lester, and E. Wilkinson, "A Low-Overhead Method for Software Process Appraisal," *Software Process Improvement and Practice,* vol. 12, pp. 339-349, 2007.

[22] C.G. Von Wangenheim, T. Varkoi, and C.F. Salviano, "Standard Based Software Process Assessments in Small Companies," *Software Process: Improvement and Practice,* vol. 11, no. 3, pp. 329-335, 2006.

[23] S. Beecham, T. Hall, and A. Rainer, "Defining a Requirements Process Improvement Model," *Software Quality J.,* vol. 13, no. 3, pp. 247-279, 2005.

[24] D. Damian, D. Zowghi, L. Vaidyanathasamy, and Y. Pal, "An Industrial Case Study of Immediate Benefits of Requirements Engineering Process Improvement at the Australian Center for Unisys Software," *Empirical Software Eng.,* vol. 9, nos. 1/2, pp. 45-75, 2004.

[25] K. El Emam and N.H. Madhavji, "Measuring the Success of Requirements Engineering Processes," *Proc. IEEE Int'l Symp. Requirements Eng.,* pp. 204-211, 1995.

[26] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide.* John Wiley & Sons, 1997.

[27] L. Mathiassen, "Collaborative Practice Research," *Information Technology & People*, vol. 15, no. 4, pp. 321-345, 2002.

[28] R.K. Yin, *Case Study Research: Design and Methods*, third ed., vol. 5. Sage, 2003.

[29] K. El Emam and A. Birk, "Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability," *IEEE Trans. Software Eng.*, vol. 26, no. 6, pp. 541-566, June 2000.

[30] J.-M. Simon, "SPICE: Overview for Software Process Improvement," *J. Systems Architecture*, vol. 42, no. 8, p. 633, 1996.

[31] V. Basili and H. Rombach, "The Tame Project: Towards Improvement-Oriented Software Environments," *IEEE Trans. Software Eng.*, vol. 14, no. 6, pp. 758-773, June 1988.

[32] M. Daneva, "Lessons Learnt from Five Years of Experience in ERP Requirements Engineering," *Proc. 11th IEEE Int'l Conf. Requirements Eng.*, pp. 45-54, 2003.

[33] M. Daneva, "Using Maturity Assessments to Understand the ERP Requirements Engineering Process," *Proc. IEEE Joint Int'l Conf. Requirements Eng.*, pp. 255-262, 2002.

[34] M. Niazi, "An Instrument for Measuring the Maturity of Requirements Engineering Process," *Proc. Sixth Int'l Conf. Product Focused Software Process Improvement*, 2005.

[35] M. Kauppinen, T. Aaltio, and S. Kujala, "Lessons Learned from Applying the Requirements Engineering Good Practice Guide for Process Improvement," *Proc. Seventh Int'l Conf. Software Quality*, 2002.

[36] M. Kauppinen, M. Vartiainen, J. Kontio, S. Kujala, and R. Sulonen, "Implementing Requirements Engineering Processes throughout Organizations: Success Factors and Challenges," *Information and Software Technology*, vol. 46, no. 14, pp. 937-953, 2004.

[37] I. Sommerville and J. Ransom, "An Empirical Study of Industrial Requirements Engineering Process Assessment and Improvement," *ACM Trans. Software Eng. and Methodology*, vol. 14, no. 1, pp. 85-117, 2005.

[38] K. Lyytinen, L. Mathiassen, and J. Ropponen, "Attention Shaping and Software Risk—A Categorical Analysis of Four Classical Risk Management Approaches," *Information Systems Research*, vol. 9, no. 3, pp. 233-255, 1998.

[39] V. Kasi, M. Keil, L. Mathiassen, and K. Pedersen, "The Post Mortem Paradox: A Delphi Study of IT Specialist Perceptions," *European J. Information Systems*, vol. 17, pp. 62-78, 2008.

[40] M. Tiedeman, "Post-Mortems—Methodology and Experiences," *IEEE J. Selected Areas in Comm.*, vol. 8, no. 2, pp. 176-180, Feb. 1990.

[41] B. McFeeley, "IDEAL: A User's Guide for Software Process Improvement," CMU/SEI-96-HB-001, Software Eng. Inst., Carnegie Mellon Univ., 1996.

[42] G. Susman and R. Evered, "An Assessment of the Scientific Merits of Action Research," *Administrative Science Quarterly*, vol. 23, no. 4, pp. 582-603, 1978.

[43] B. Kitchenham, S. Pfleeger, D. Hoaglin, K.E. Emam, and J. Rosenberg, "Preliminary Guidelines for Empirical Research in Software Engineering," *IEEE Trans. Software Eng.*, vol. 28, no. 8, pp. 721-734, Aug. 2002.

[44] R. Rapoport, "Three Dilemmas in Action Research," *Human Relations*, vol. 23, no. 6, pp. 499-513, 1970.

[45] R. Baskerville and T. Wood-Harper, "Diversity in Information Systems Action Research Methods," *European J. Information Systems*, vol. 7, no. 2, pp. 90-107, 1998.

[46] A.M. Pettigrew, "Longitudinal Field Research on Change: Theory and Practice," *Organization Science*, vol. 1, no. 3, pp. 267-292, 1990.

[47] M.B. Miles and A.M. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook.* Sage Publications, 1994.

[48] E.A. Weitzman and M.B. Miles, *Computer Programs for Qualitative Data Analysis.* Sage Publications, 1995.

[49] Y. Lincoln and E. Guba, *Naturalistic Inquiry.* Sage, 1985.

[50] J. Mason, *Qualitative Researching*, second ed. Sage, 2002.

[51] C.B. Seaman, "Qualitative Methods in Empirical Studies of Software Engineering," *IEEE Trans. Software Eng.*, vol. 25, no. 4, pp. 557-572, July/Aug. 1999.

[52] N.P. Napier, "Improving Practices in a Small Software Firm: An Ambidextrous Perspective," doctoral dissertation, Computer Information Systems, Georgia State Univ., 2007.

[53] M. Kauppinen, S. Kujala, T. Aaltio, and L. Lehtola, "Introducing Requirements Engineering: How to Make a Cultural Change Happen in Practice," *Proc. Joint Int'l Conf. Requirements Eng.*, pp. 43-51, 2002.

[54] B.W. Boehm, "Get Ready for Agile Methods, with Care," *Computer*, vol. 35, no. 1, pp. 64-69, Jan. 2002.

[55] B.W. Boehm and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed.* Addison-Wesley, 2004.

[56] P. Checkland and J. Scholes, *Soft Systems Methodology: A 30-Year Retrospective.* John Wiley, 1999.

Nannette P. Napier received the bachelor's degrees in computer science from Spelman College and in computer engineering from the Rochester Institute of Technology, and the doctorate degree in computer information systems from the J. Mack Robinson College of Business of Georgia State University. She is currently an assistant professor of information technology at Georgia Gwinnett College. Prior to attending graduate school, she worked professionally as a senior software engineer for eight years. She engages in action research that addresses challenges of software development firms such as effectively managing software projects, creating and managing software requirements, and using agile development methodologies. Her research has been published in several journals, including *Information Systems Journal* and *Software Process: Improvement and Practice.* Her research has also been presented at refereed conferences, including the International Conference on Information Systems (ICIS), America's Conference on Information Systems (AMCIS), Hawaii International Conference on System Sciences (HICSS), and Southern Association for Information Systems (SAIS). She is a member of the Association for Information Systems.

Lars Mathiassen received the master's degree in computer science from Aarhus University, Denmark, in 1975, the PhD degree in informatics from Oslo University, Norway, in 1981, and the Dr Techn degree in software engineering from Aalborg University, Denmark, in 1998. He is currently a professor of computer information systems at Georgia State University, Atlanta. His research interests include information systems and software engineering, with a particular emphasis on process innovation. He is the coauthor of *Professional Systems Development* (Prentice-Hall, 1990), *Computers in Context* (Blackwell, 1993), *Object Oriented Analysis & Design* (Marko Publishing, 2000), and *Improving Software Organizations* (Addison-Wesley, 2002). He is a member of the IEEE, the ACM, and the Association for Information Systems.

Roy D. Johnson received the bachelor of science degree in health, physical education, recreation, and dance in 1970, the master of arts degree in physical education and art in 1971 from Appalachian State University, and the doctorate of philosophy degree in anatomy, art, and dance in 1980 from the University of Oregon, with two postdocs in management information systems from the University of Minnesota (1986) and Indiana University (1988). He is currently an extraordinary professor at the University of Pretoria, South Africa. His academic career includes employment at Bowling Green State University, Roosevelt University, Nanyang Technological University (Singapore), University of Aarhus (Denmark), DePaul University, Portland State University, Idaho State University, Florida Atlantic University, and Georgia State University. His research has been published in multiple refereed journals and presented at multiple refereed conferences, including the International Conference on Information Systems (ICIS), America's Conference on Information Systems (AMCIS), European Conference on Information Systems (ECIS), Pacific Asia Conference on Information Systems (ECIS), and Southern Association for Information Systems (SAIS). He has won six Best Paper awards, more than $750,000 in grants, and university teaching awards and has numerous refereed journal articles and two books to his credit. He engages in research that addresses challenges in the areas of systems analysis and design, project management, as well as IS education and assessment. He is a member of the Association for Information System, serving previously as the VP of Education and currently as the VP of Accreditation.