

# Relating Reflection Workshop Results with Team Goals

Sylvia Kopczyńska

*Poznan University of Technology, Institute of Computing Science  
Piotrowo 2, 60-965 Poznań, Poland  
E-mail: sylvia.kopczynska@cs.put.poznan.pl*

Received: 31 May 2014; revised: 17 October 2014; accepted: 06 November 2014; published online: 10 January 2015

**Abstract:** Agile software development promotes constant learning from experience. Consequently, the feedback of project team members helps to improve and adapt to new future challenges. A reflection workshop is one of the methods that aim to introduce this concept. However, there still exists the question of how to handle reflection results so that the gathered feedback is addressed. In this paper we present our approach, called RefGoal. It aims to treat the results of reflection workshops in a systematic manner. The core of our idea is to align these results with the goals of the organization in which the project is being run. Thereafter, we report the results of our exploratory case study. Its goal was to preliminarily evaluate feasibility of the proposed idea in the context of three software development teams working on tailor-made web applications. The results show that the cost of using RefGoal for reflection workshops is about the same as that of a typical workshop (at max. 1.5 h), and that there exists a considerable number of insights that lead to valuable updates of team goals. Based on the results, we also uncovered some open questions, regarding handling reflection workshop results, which require addressing in the near future.

**Key words:** reflection workshop, retrospective, GQM + Strategies, project goals, agile

---

## I. INTRODUCTION

Improvement in agile software development projects is about continuous or continual improvement, which requires learning from experience, from previously taken action. Applying this idea means stopping for a moment, at some point, and reflecting on practice. In order to draw conclusions, a careful analysis of past situations is needed, as well as comparing it with the desired and the current state. Deliberating various alternatives and taking diverse perspectives is vital. Reflection (also called retrospection) can be used as a tool for continuous learning organizations. The context, including the participants, in which reflection is to be fostered determines its shape. It is vital to be aware of the time perspective, making sure that the conclusions are up-to-date. For example, in an agile software project that has short iteration cycles, the results of reflection are employed in the next iteration or in the next several stages. So, it is important that the benefits of the reflection insights are not approached merely in a short-term

perceptive, but that they are combined with a more extensive mechanism for avoiding similar future errors and intensifying good practices in equivalent situations. Hence, the applied reflection technique must be well-integrated with an organization's decision-making procedures. There are many great works on how to organize a reflection workshop, as well as about various specific techniques that can be used (e.g., [1]), but there is a scarcity of information on how to systematically handle the output. For example, should a team hang the results on a wall or create a set of action items? Making use of the results is the key to achieving improvement in an organization, informed by the lessons learned from experience. Lack of handling the reflection's results leads in short order to the problem of team dissatisfaction, which is frequently mentioned by agile software practitioners. In both short- and long-term perspective, this leads to a loss of money and resources.

Our idea, called RefGoals (presented in Section III), is to include the results of a software development project retro-

spection with the goals of the organization in which the given project is being run. The first concept it originates from is learning from experience by applying a lightweight reflection technique with the use of the starfish technique, which we shortly present in Section II. 1. The second concept is expressing the organization's business model and decision-making mechanism with a goal-based approach, using the GQM + Strategies method (GQM + S), which we introduce in Section II. 2. Combining the two concepts allows to obtain a rationale for the introduction of certain improvements and a commitment to their execution. Although we believe that the concept might be applied organization-wide, we focus on reflecting on practice by a team and relating it to all the goals they are involved in. In order to investigate the feasibility of our idea, we performed an early evaluation in the form of a case study (the design is presented in Section IV). We discussed the results from three software development projects, regarding cost, ease of use, and the extent to which the effects of a reflection workshop can be integrated with team goals, in Section V. We positioned our work among others regarding the reflection method in Section VI. The lessons we have learned and the identified open questions look promising and draw future work directions, which we described in Section VII. The main contributions of the paper are as follows: (1) we present our idea for combining the results of reflection workshops with team goals, which we call RefGoal; (2) we report the lessons we have learned in a preliminary study regarding the cost and ease of use of the proposed idea; (3) we discuss our findings on the types of reflection workshop results and on the extent to which they can be integrated with team goals.

## II. BACKGROUND

### II. 1. Reflection Workshop

Reflection on practice in a software project, also called a retrospective, is a technique used to identify and document good and bad practices, actions or behaviors, and to set new goals for subsequent stages. It allows for learning from experience. One of the first to include reflection ideas as a required component of professional learning was Donald Schön [2]. In his book, he suggested a three step approach: first learn in a systematic way, then reflect, and finally take appropriate actions. This idea is called reflection-on-action.

Later on, the idea of reflection-in-action was introduced, i.e., a more advanced ability which assumes that, while performing, practitioners always relate the problem they are facing with their knowledge, emotions, and experience, and this allows them to make the best possible decisions.

The way we perform reflections is closely related to the type of improvement strategy used in an organization.

Continuous improvement assumes literally continuous effort to make the processes, products, services and environ-

ment better; while continual means introducing improvement incrementally, by discrete steps.

The latter option usually requires that reflection analyze a longer time period, which in turn determines the applied techniques.

Kent Beck stated that running a project is not just about driving in the right direction, but also about taking corrective action based on constant observation of the entire surroundings [3]. Hence, in agile software development projects reflection is an essential element of the project life-cycle, and can be performed in various ways. The book by Derby and Larsen [1] clearly presents a common process of doing reflection in workshop form, and provides a handful of valuable good practices. The process is composed of five consequent steps that all team members take part in. First, the stage must be set (Set the Stage), which means the workshop should be opened, the rules and goal presented. The second step, Gather Data, is about performing a reminder of what happened, as objectively as possible, during the time period that is being reflected on. Then, the team considers the data in order to identify strong points and issues, which is called Generate Insights. The fourth step, called Decide What to Do, is about finding solutions to the identified insights, e.g., improvement actions. The last step is to summarize and conclude the workshop, i.e., to Close the Retrospective.

There exists a number of techniques that can be used to execute each step of a reflection workshop, e.g., gathering data using Triple Nickels or Color Code Dots, generating insights using Five Whys or Learning Matrix. Identifying good and bad points, and deciding if they are worth pursuing, can be done with the use of the Starfish technique. In that case all team members are to generate insights in five categories: keep doing (all that they want to preserve), less of, more of (all that should be increased or decreased in intensity), start/stop doing (all that they would like to introduce to or remove from the process).

### II. 2. GQM + Strategies

GQM + Strategies [4] is a method for aligning, integrating, and communicating goals, strategies and measurement programs across an organization. This way an organization can model their values, express their business drivers, and communicate what they want to achieve. The method encourages conveying the business of the organization with a hierarchy (a net) of goals and strategies – the activities that fulfill the goals. A general example of such a hierarchy, called a grid, is presented in Fig. 1. A decision on approving a certain goal or selecting a proposed strategy should be based either on facts – context factors (CF), or on assumptions (A). Business objectives are to be measurable, so each goal should be connected with a GQM graph (measurement goals, questions and metrics related to one another). As a result of applying GQM + Strategies, we get a measurement program which presents business objectives (drivers, values) of an organization in the form of a grid, i.e., a set of related elements:

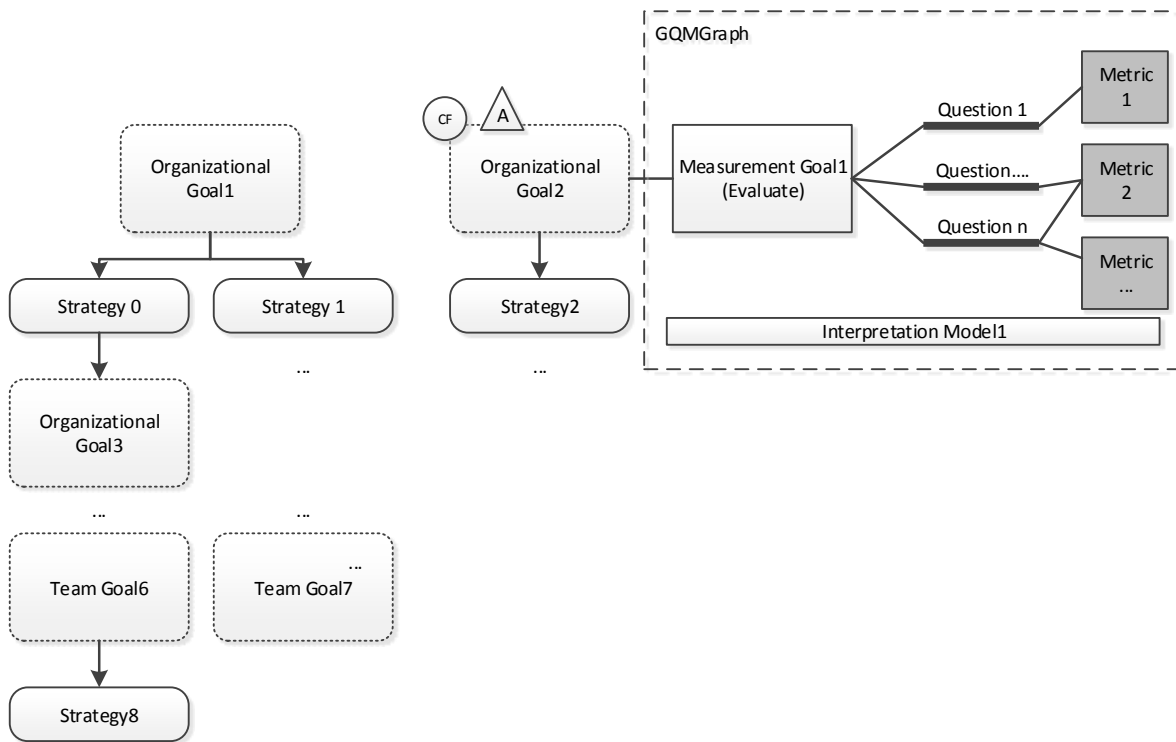


Fig. 1. An example of the GQM+ Strategies grid

goals, strategies, important environmental factors, and GQM graphs.

Usually goals and strategies are defined in a top-down manner, i.e., from the business level, through department and team goals, to team and project level goals. As a result, each team can see their link with and impact on the higher level goals, e.g., the relation between the decision to increase test coverage by 10% on the team level, and its relationship to the goal of achieving higher client satisfaction. Maintenance of GQM+ Strategies grids is executed according to Experience Factory and the QIP cycle [5]. All GQM+ Strategies elements are stored in an experience database and could be easily retrieved to improve future actions.

### III. RefGoal

Our idea, i.e., the method we call RefGoal, is about holding a workshop that captures experience, and handling the results by using the notation of GQM+ Strategies. In this way it aims to: (1) reflect on practice to learn from experience; (2) provide forward feedback; and (3) set the goals and strategies for improvement. Simultaneously, a measurement mechanism can be added, so that a team can monitor if they are implementing their improvement ideas and evaluate the achieved state later on. Moreover, our idea not only

includes project-based learning, but promotes the organizational learning paradigm as well. The retrospection results are combined with organizational grids, which are items stored in an organizational experience database. Including the results of retrospection in the grid also means explicitly expressing the impact on organizational goals and strategies, which means showing the value of improvement-based actions on the business model.

#### III. 1. Roles

RefGoals is a three-step method – Preparation, Workshop, Analysis – and involves all team members. They play the roles of Leader, Facilitator, Recorder, and Participant. Each team member is a Participant. Then, the participant responsible for an initial presentation of the project period or practice to be reflected upon plays the role of Leader. The Leader role could be played by a project manager, team leader, or any other team member. The person is required to have access to the products and measurement data of the project; it is recommended that the role is played in a round-robin manner. Further, a Facilitator is a person who moderates the Workshop, takes care of the pace, progress and achieving the workshop's goal. Finally, a Recorder is a participant responsible for documenting the results of the meeting. The roles should be assigned to team members that have worked on the same piece of work (e.g. increment, sprint). Including the representatives of the client or end-user might be beneficial,

e.g., when there are problems in cooperating with a client. However, the Facilitator should first analyze who would be crucial for achieving the goal of the meeting. Certainly, in one project multiple Workshops could be organized, for example, one with the internal team, then one with the client, or just one with all the stakeholders.

### III. 2. Steps

The steps of RefGoal are the following:

- I. **Preparation** – the Facilitator is required to, organize the Workshop (e.g. schedule the workshop, send invitations) and assign the roles to the Workshop’s participants. The Leader prepares an introductory presentation which relates the period to be reflected upon, its goals, and results, including the results of measurement.
- II. **Workshop** – the step is the cornerstone of this method. It implements the framework of a reflection workshop proposed by Derby and Larsen [1], and combines it with the starfish technique and the GQM + Strategies method. The agenda of the workshop is designed as follows:
  1. **Set the Stage.** At first, the Facilitator should begin the workshop by welcoming the participants and reminding them of the rules of the workshop and the expected duration. Simultaneously, the Recorder should start recording, e.g., turn on a voice-recorder or a camera device.
  2. **Gather data.** Secondly, the Leader, using the prepared presentation, recalls the iteration/stage to the team members, as well as its goals, progress and results.
  3. **Generate insights.** Thirdly, the Facilitator distributes post-its and the Participants individually document their feelings, ideas and experience: what they liked and what they did not like. Participants are also expected to indicate if these are positive or negative findings.
  4. **Generate insights.** Then, all the participants stick the post-its onto a board; they clarify, analyze, and stack the same types of observations together.
  5. **Decide what to do.** Next, participants propose solutions to the identified insights. They stick the post-its to a starfish diagram, meaning that they divide them into five categories: keep doing, less of, more of, start doing, stop doing, depending on the type of agreed solution. The solutions would be: tasks or goals, such as an improvement or a request to remain in a certain state. The goals shall be defined by the team (e.g., the magnitude and the time-frame for a goal), and if possible, a measurement mechanism (GQM graph) shall be proposed. If there exists a set of team goals in the form of GQM + Strategies grid, then it is

recommended to augment the grid at the same time.

6. **Close the retrospective.** The workshop should be finished by the Facilitator summing up the results and thanking everyone for their participation.

III. **Analysis** – the Leader turns the recordings of the group work into notes, action items, and measurable organizational goals and strategies. Then, he provides these results to the Participants for their acceptance. Some minor Participants’ improvement suggestions, e.g., typos, grammar errors, should be incorporated into the results. If there occur any major change requests, another workshop should be organized to solve the issues together, in a team, as the result is a set of team goals.

At the beginning of the Workshop, an introductory warm-up exercise could be done to encourage creativity and embolden Participants to share their thoughts, as suggested in [1]. Such a retrospection workshop can be organized after each iteration (sprint or stage).

### III. 3. Input and Output Products

The items used in the method are the following:

- input – current measurable goals and strategies, measurement data and its interpretation, iteration plans, as well as other project data.
- an introductory presentation – a presentation given by Presenter reminding Workshop Participants about the iteration progress, goals, results.
- workshop rules and policies – information about rules of Workshop, including tasks, expectations, responsibilities, agenda, is presented to Participants at the beginning of the Workshop.
- workshop recordings – all data recorded during the Workshop, including an introductory presentation, workshop rules and policies, ideas, problems and solutions generated during the Workshop, etc.
- measurable organizational goals and strategies – the recordings of the Workshop, formalized using the GQM + Strategies method, the goals and strategies that are expected to improve the current situation.

### III. 4. Including Reflection Results into a Grid

The main results of a reflection workshop are: (1) insights – the identified symptoms of problems or good practices, or the problems and good practices themselves; (2) solutions – proposals on how to solve the problems, polish or enforce the good practices. One solution could solve many insights, and one insight can be an impulse to introduce multiple solutions. Depending on the type of an insight-solution pair, particular action will be taken. However, a relation with team goals should always be demonstrated. So far we have identified the following types of insight-solution pairs (also including the results of the study described in Section V):

- **TASK** – Some problems require performing a simple task, e.g., improving workspace quality through

repairing an air conditioner. Such insights could be documented in a task management tool, and have a person assigned to be responsible for completing the task. Then, finally, during the next reflection workshop, it should be verified whether the task was completed and if it solved the identified problem.

- **GOAL OR STRATEGY** – Some reflection results describe a certain state that a team would like to preserve or achieve. In such situations, we propose to include the results to their GQM + Strategies grid as follows. The gathered insights are the rationale, so they are either context factors (e.g., data, known issue) or assumptions (i.e., we believe the problem or good practice has such effects, results, and symptoms). The organizational goal should express the state desired by the team. The means of improving the current situation should be strategies of achieving the organizational goal. Sometimes no new goal is identified, but only a new strategy. A model of mapping the results to GQM + Strategies grid is presented in Fig. 2, and an example in Fig. 3.
- **DECISION** – There are some insights that require making decisions, e.g., insight = it is too cold in the room, solution = the temperature should be set for 21. The decision could regard the way the team works, which we call DECISION-RULE, or might regard change (e.g., a new template for documenting requirements) which we call DECISION-CHANGE. The decisions usually do not have the nature of a task, sometimes they can lead to introducing a new goal or strategy. But it is treated separately, as the team does not decide to include it in their set of goals.

- **AFFIRMATION** – There are some insights that express an opinion on the current state of the team, e.g., great atmosphere, we did a good job with eliciting requirements, but having analyzed them, the team decided not to set it as their goal. The role of such statements is to . Such results can be included in an experience factory – database of lessons learned.

There exists an open question: *are there any other types of insight-solution pairs?*

#### IV. EARLY EMPIRICAL VALIDATION – DESIGN

In our study we wanted to investigate and better understand the RefGoal method in real-life settings. Hence, we decided to carry out an exploratory case study. This type of approach is useful for exploring a phenomenon thoroughly and demonstrating its quality in use[6]. The goal of the case study was to characterize and understand RefGoal, focusing on its cost, ease of use, and feasibility from the point of view of someone interested in applying the method to agile software development. We refined our objective into the following set of research questions:

- RQ1:** What is the cost: the effort, complexity of activities and what resources are used to execute them?
- RQ2:** Do the participants perceive the method as difficult?
- RQ3:** To what extent can the results of a reflection workshop be included in a GQM + Strategies grid?

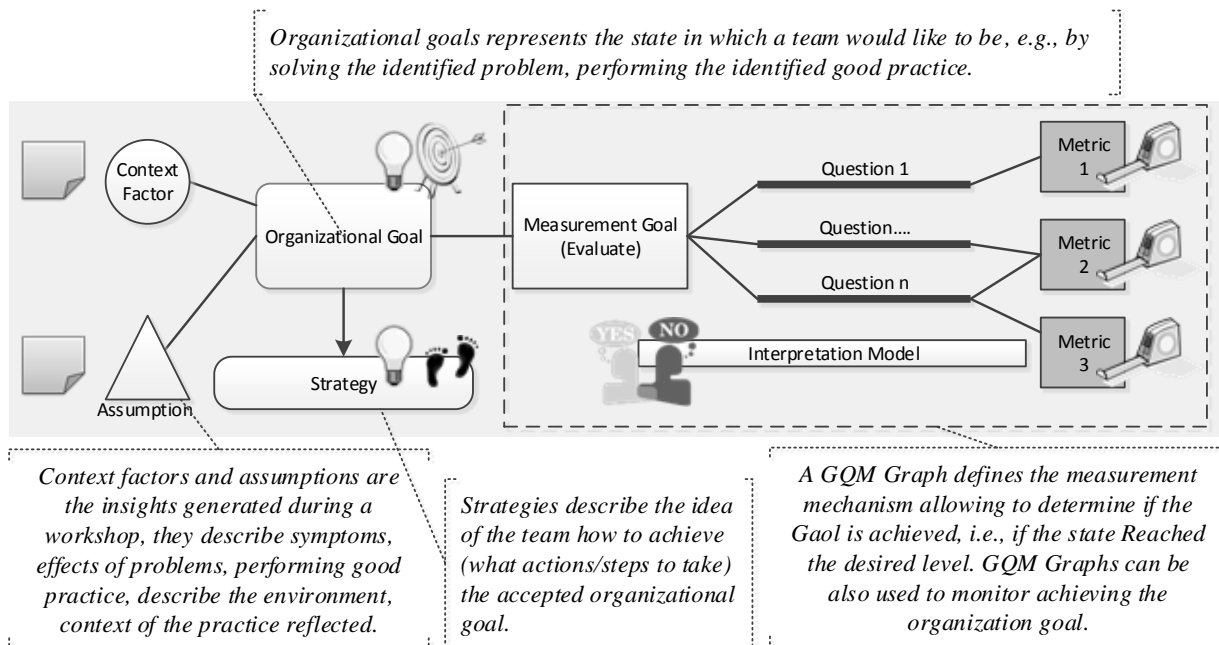


Fig. 2. Mapping the results of a reflection workshop to GQM + Strategies elements

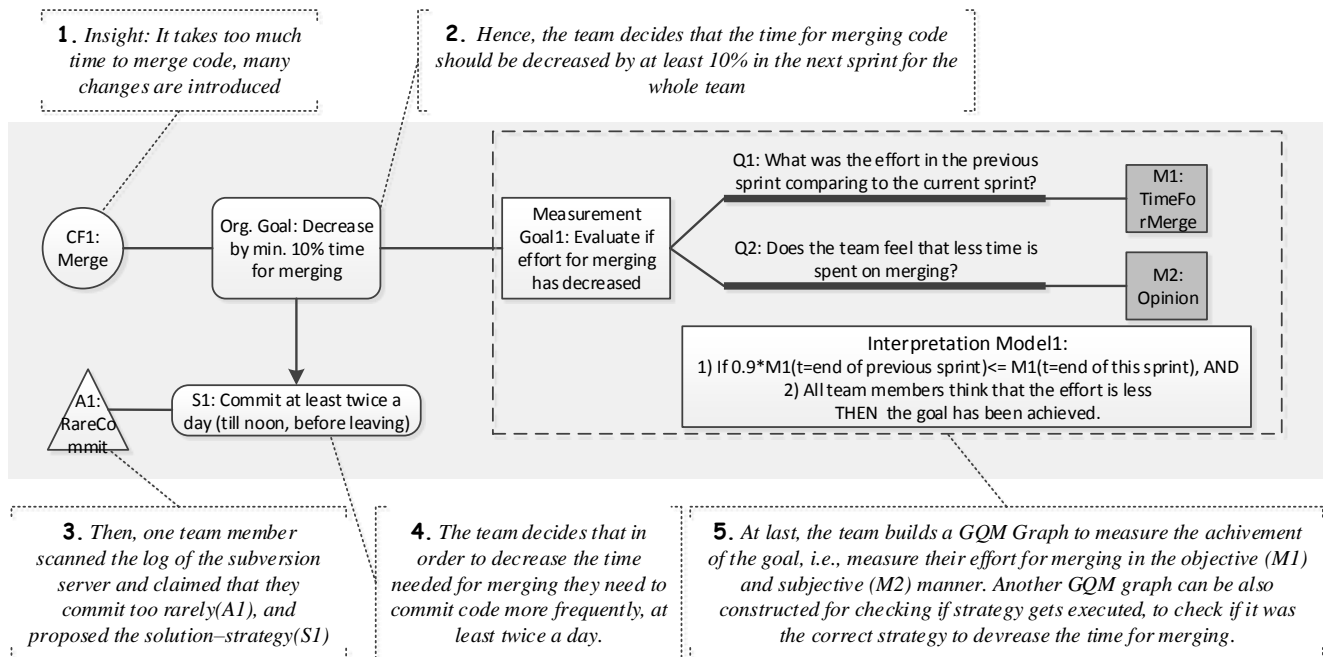


Fig. 3. An example GQM + Strategies grid with results of a reflection workshop

#### IV. 1. Case and Subjects Selection

The projects that we selected for our case study were three projects carried out in 2013/2014 (#2) and 2012/2013(#1) at Poznan University of Technology (PUT) (the teams are described in Table 1). At PUT the Software Department is responsible for software used by various units, from accounting to research and education. This Department cooperates with the Infrastructure Department, responsible for the management of hardware and network infrastructure (e.g., servers), and the Department of Service Development, responsible for the quality of software services provided at the University (e.g., Service Desk). Moreover, the Software Department closely works with the Software Development Studio, which makes it possible to increase its software development capacity. Each year within this cooperation ca. 5 projects are run. In each project, the following people take part: employees of the Software Department (responsible for product quality), one person from the Software Engineering Laboratory (responsible for process quality, so-called quality assurance), customer representative(s) (University employee(s) responsible for business value and the management of the project budget), representatives of end-users (University employees, students or University partners who are to use the software product), two or three Master students (who play the roles of Project Manager, Analyst and Architect), and four Bachelor students (who are software developers and testers). Hence, each team is composed of senior and junior technical and process experts, as is the case in other software development

organizations (e.g., industry). Each project is run according to the XPrince methodology [7] or PRINCE2 + Scrum, and lasts for a year (they work ca. 2-2.5 days per week). There are two or three releases, and each of them is divided into at least two increments. The teams have dedicated offices equipped with hardware and software. Moreover, each project's budget allocates a salary for each team member, which is at the level of salaries that is possible for students to earn while working part-time in industry. More information about the environment can be found in the paper by Kopczyńska, Ochodek, Nawrocki [8]. The Project Manager and Analyst of each team was to firstly take part in a short tutorial given by the author, and to use the method; afterwards, they were to use the method in their team after each increment or sprint. Each team received the same guidelines and no learned lessons were incorporated during the study.

#### IV. 2. Data Collection and Analysis Procedures

In this exploratory case study we considered the following procedures for data collection feasible:

- Surveys – a short questionnaire distributed right after a workshop step, asking the participants for their impressions about the activities and products.
- Output products – we collected the output products: insights, solutions created by the participants during workshops.
- GQM + Strategies Grids – we collected the grids exposing the goals of each team.
- Interview recordings – we managed to conduct a struc-

- tured interview with one team and record an audio file.
- Researcher observations – we took notes during the study to record important information.
- Communication with team participants via mail.

For analyzing the qualitative data we used the guidelines provided by Charmaz [9]. As such we performed initial line coding, coding and axial coding. During coding and constant comparison, we extracted codes for different types of reflection workshops results (insights and solutions).

### IV. 3. Validity Procedure

Although we did not aim to find any cause-effect relationships, we addressed internal validity by using the triangulation of data collection sources. Our case study was not designed to provide statistically valid conclusions for all software development projects or organizations. Its aim was to thoroughly describe and deeply understand the proposed method within its context. We tried to provide an in-detail description of the context in Section IV. Although there exist different types of Software Studios, which execute the goals of conscious application of good software engineering practices in different manners, we claim that our Software Development Studio adequately simulates the real environment. There are real investors, user representatives, dedicated workspaces, the teams work with other employees, and most importantly, they sign contracts, may earn a salary equal to those available on the market, and participate in transition. Because our study lasted for a year (as each project's duration), and RefGoal was implemented several times by each team, it was possible for a learning effect of the method to people. RefGoal is a workshop which might be influenced by many environmental factors (e.g., light or temperature in the room, placement of seats). We did not try to mitigate the factors, as we wanted to investigate the method in a real-life setting, therefore in a context in which it is influenced by such factors. However, we asked team members a general question each week about any obstacles and impressions from the performed tasks. We also analyzed the results of the reflection workshop from this perspective.

We addressed the reliability by describing the RefGoal method. To make our study reliable, we linked all the data to the project's repositories, developed an interview guide, and documented the data collection process. There is also a risk, regarding construct validity, that the results will be influenced by the research's expectations. Although all team members were required to execute RefGoal (the requirements of one of their courses), they were expected and encouraged to freely express their opinions.

## V. RESULTS

### V. 1. Cost

Each team took part in a 45-minutes-long initial tutorial presenting the idea of reflection and the guidelines of performing RefGoal.

Team A declared that applying the method took them ca. 15.5 h, specifically: 3 h for an initial presentation preparation by Leader (Project Manager); 6 · 1.25 h for the Workshop and 5 h for Analysis. Similarly, for the workshops of Team B: the first one took 1.5 h, the next three 1.0 h, and they involved all team members. Team C did not note down the duration of the workshops. All teams invited only the development team, project managers, analysts and architects for the reflection workshops.

The resources required using post-its (at max. 10 per person) and a pen. In the case of Team A, an overhead projector was used, as well as a presentation of the project's progress.

*Conclusion: The cost and resources of using RefGoal are similar to other types of workshops (e.g., [10]) and took max. 1.5 h. The only special requirement are post-its.*

### V. 2. Ease of Use

All team members agreed, after the initial tutorial, that they were ready to hold a RefGoal workshop on their own, without any further help. During the project status meetings, which were held once a week, no team identified any problems with this approach to reflection workshops. One team (Team B), after the first workshop, asked questions regarding including results of reflection with their grid. The questions were: "Should all insights and solutions become goals?", and "Could you repeat how to map the results of reflection to the grid?".

*Conclusion: No obstacles were identified that could hinder using the method. The recommendation might be to very clearly explain mapping insights to the elements of GQM + Strategy grid.*

### V. 3. Relation of Reflection Results with Team Goals

In order to investigate the feasibility of the method, we analyzed the reflection workshop results and compared them with the GQM + S grids of each team.

**Team A.** The workshop of Team A generated 31 ideas: 14 positive and 16 negative. Two positive findings were mentioned several times, i.e., idea 25 – 4 times; idea 24 – 2 times; one negative finding was mentioned 3 times, and the others were unique. Four team members judged the workshop as a good technique, providing good basis for further project improvement. In the Analysis step of RefGoal, Team A decided to introduce two organizational goals into their grid, as they were the most important and had the highest impact for them. The majority of positive findings were about the culture of collaboration and the way they worked together. During the Workshop, team members confirmed that it was essential, and all of them wanted to continue this style of working. One of the goals was connected with keeping the project knowledge base up-to-date by maintaining the system that stores it. The second goal was related to making requirements more consistent, which generated a strategy, for the Analyst, of continuous requirement analysis. The semi-structured interview revealed that the goals added after the Workshop were

Tab. 1. Descriptions of the teams that participated in the study

Description	Use of RefGoal
<p><b>Team A</b> developed a web application for the management of organizational tasks of University employees. Team members included: 4 developers, 1 analyst, 1 project manager, 1 user representative, 1 executive, 1 future maintainer. Methodology: XPrince.</p>	<p>Training just before using the method (45min). Once mid-development, after the first release, before starting the second one.</p>
<p><b>Team B</b> developed a web application for collecting data and generating reports regarding professional careers of University graduates. Team members included: 4 developers, 1 analyst, 1 project manager, 2 user representatives, 1 executive, 1 future maintainer, 1 quality assurance. Methodology: XPrince + Scrum.</p>	<p>Training just before using the method (45min). The method was used 4 times after each sprint.</p>
<p><b>Team C</b> developed a web application for management of the assignment to Master and Bachelor theses. Team members included: 4 developers, 1 analyst, 1 project manager, 2 user representatives, 1 executive, 1 future maintainer, 1 quality assurance. Methodology: XPrince.</p>	<p>Training just before using the method (45min). The method was used 3 times after each release.</p>

regarded as “most influential” and “allowed us to improve the software development”.

**Team B.** During 4 sprints, the team generated a total of 51 insights (17, 17, 4, 3 correspondingly), 17 of which were positive (5, 4, 3, 5, correspondingly). A great number of them (20) regarded teamwork aspects, further 15 insights considered process-related issues. In the first two sprints, the insights related to the techniques and methods used to develop the products. The insights were turned into 27 solutions: 5 tasks, 6 goals and 4 hidden goals (not added to the grid), 2 strategies, 10 decisions. Only the goals derived from negative insights were added to the grid. The team explained that these were the things they focused on, and treated them as sharpening a saw or solving a problem. For example, there was a problem with daily scrum meetings. The first defined goal was to begin them at a specific time and in the way the team believed was best suited for them (a kind of checklist), and for two sprints the team was trying to achieve this goal. However, during the last observed sprint, the team spotted that again there was a problem with their Daily Scrums. The root cause was the end-of-semester time, which means that for students many additional tasks crop up alongside project-related tasks. So, they decided to introduce a strategy to help achieve the goal – a penalty for the team member who spoils the meeting. The team members were proud of themselves when they saw that the reflection workshops were improving (a GQM graph was used to monitor the time and effectiveness of the workshops).

**Team C.** The team generated 31 insights during 3 workshops (5, 19, 8). There were 11 positive insights and 9 of them were about teamwork. The negative insights regarded mainly external factors (3) (e.g., long response time from the Service Development Department) or environmental factors (10) (e.g., a lack of office supplies). 9 solutions were proposed: 3 strategies, 3 decision-rules of new behavior (e.g., including additional time for communication in the estimation of tasks), 2 goals and 1 task. The goals regarding positive aspects of teamwork were not included into their grid, although team

members defined good communication and a well-motivated team as their key success factors.

*Conclusion:* There was a considerable number of results of reflection workshops that could be integrated with the GQM + S grid. Teams included goals derived from negative rather than from positive insights.

## VI. RELATED WORK

Generally, software process improvement methods can be assigned to one of two categories: inductive (problem-based) and prescriptive (model-based) [11]. The model-based methods recommend applying some best practices, while the problem-based methods consider existing problems and weaknesses as the driving force for improvement [11]. The latter approach is required and implemented in many project management methodologies, e.g., by recording and storing experience. For example, PRINCE2 [12] regards it essential to first maintain a lessons learned log, then to include the experience in project reports, and finally to pass this experience onto later projects. Agile methodologies shifted the focus of the problem-oriented approach, so the problems are identified based on the feedback of project team members. As a consequence, stand-up meetings and reflection workshop methods were introduced. As the results of a reflection workshop, i.e., the output, we get new perspectives on experience, changes in actions and behaviors, new steps to apply, and the confirmation of readiness towards application and commitment. Although many researchers discuss the idea of reflection quite comprehensively, e.g., [13], still, there is a scarcity of empirical evaluation of reflection on practice, especially in software development settings. Some researchers proposed other methods, e.g., workshops introduced by Dingsøyr and Hanssen [14]. They presented a technique to be used after each iteration, similarly to Salo et al. [15], who recommended



performing a post-iteration workshop for projects adopting Extreme Programming (XP). Collier et al. [16] defined a process for project post-mortem reviews. In the literature we can also find some lessons learned from applying various methods, e.g., Fajtak [17] reports their experience from using kick-off meetings and retrospectives by a division of Siemens AG Austria. Dingsøy [18] compared several techniques used in software development. Their experience shows the positive impact of retrospectives, and that it is important to: ensure good facilitation of a workshop, an atmosphere that will cause the participants to reveal honest impressions about the practice, and to document the results, so that they can be set and monitored.

Usually, some creativity engaging and social techniques must be used during a reflection workshop. For example, brainstorming, writing down ideas on pieces of paper, collaboratively creating a list of ideas through each member adding their own idea. Moreover, the applied techniques are used to analyze root causes of identified problems, such as Ishikawa diagrams or, more advanced, combining creativity and root cause analysis techniques [19].

Although a framework for reflection workshops was clearly defined by Derby and Larsen [1], the consequent steps (phases) can be executed in many different ways, and various factors can influence the reflection process. Accordingly, some researchers focus on specific aspects or steps. For example, reflection on practice may be a particularly huge challenge in big organizations or/and for distributed teams. Schmmmer et al. [20] spotted this problem and designed a specific process supported with a software tool. Margaritas et al. [21] developed a collaborative software tool. They recommend graphically visualizing the results, and describing the experience of using the tool by a distributed team. The problem with executing certain steps of a reflection workshop was spotted by Bjarnason et al. [22]. They proposed the timeline technique and evaluated it in two different projects. A great majority of studies focuses on the techniques of performing reflection, rather than on methods of handling its results.

## VII. CONCLUSIONS

In the paper we described our idea, called RefGoal, of relating the results of a reflection workshop with the goals of a software development team. The method aims to handle the reflection findings in a systematic way. It combines two methods: a lightweight reflection workshop organized according to the guidelines of Derby and Larsen [1], with the starfish technique, and the GQM + Strategies method. Our preliminary study shows that applying our idea was only as time-consuming as good practices for a workshop state (e.g., [10]), and there were no problems identified with using it. It allowed team members to identify the areas requiring

improvement. The teams explicitly included the reflection results in the project goals, which allowed them to handle the root causes. The method and its outcomes were judged as valuable by the participants of the study. Moreover, the study allowed us to identify other types of results of reflection workshops, which require different ways of handling. Future work needs to focus on thorough empirical evaluation in more projects and organizations, e.g., in a bigger organization or for a longer project.

## Acknowledgments

I would like to thank Prof. Jerzy Nawrocki for his support and valuable guidance with regard to the solution. The author also thanks the team members of the projects who participated in the study.

## References

- [1] E. Derby, D. Larsen, *Agile Retrospectives: Making Good Teams Great*, Pragmatic Bookshelf Series Pragmatic Programmers, LLC, 2006.
- [2] D.A. Schön, *The reflective practitioner: How professionals think in action*, volume 5126, Basic books, 1983.
- [3] K. Beck, C. Andres, *Extreme Programming Explained: Embrace Change (2nd Edition)*, Addison-Wesley Professional, 2004.
- [4] V. Basili, A. Trendowicz, M. Kowalczyk, J. Heidrich, C. Seaman, J. Münch, D. Rombach, *Aligning Organizations Through Measurement*, Springer, 2014.
- [5] V. R. Basili, *The Experience Factory and its Relationship to Other Improvement Paradigms*, [In:] *Proceedings of the 4th European Software Engineering Conference on Software Engineering (ESEC '93)*.
- [6] P. Runeson, M. Host, A. Rainer, B. Regnell, *Case Study Research in Software Engineering: Guidelines and Examples*, Wiley, 2012.
- [7] J. Nawrocki, L. Olek, M. Jasinski, B. Paliświat, B. Walter, B. Pietrzak, P. Godek, *Balancing agility and discipline with XPrince*, [In:] *Proceedings of RISE 2005 Conference (in print)*, volume 3943 of LNCS, pages 266-277 Springer Verlag, 2006.
- [8] S. Kopczyńska, J. Nawrocki, M. Ochodek, *Software development studio – Bringing industrial environment to a classroom*, [In:] *Software Engineering Education based on Real-World Experiences (EduRex), 1st Internat. Workshop on*, pages 13-16 IEEE, 2012.
- [9] K. Charmax, *Constructing Grounded Theory*, SAGE Pulications, 2006.
- [10] E. Gottesdiener, *Requirements by Collaboration*, Addison-Wesley, 2002.
- [11] N.P. Napier, L. Mathiassen, R.D. Johnson, *Combining Perceptions and Prescriptions in Requirements Engineering Process Assessment: An Industrial Case Study*, IEEE Transactions on Software Engineering **35**(5), 593-606, (September 2009).
- [12] PRINCE2, *Managing successful projects with PRINCE2*, Office of Government Commerce (OGC), Stationery Office Bookssher, 2009.
- [13] P.C. Blumenfeld, E. Soloway, R.W. Marx, J.S. Krajcik, M. Guzdial, A. Palincsar, *Motivating project-based learning: Sustaining the doing, supporting the learning*, Educational Psychologist **26**(3-4), 369-398 (1991).

- [14] T. Dingsøy, G.K. Hanssen, *Extending agile methods: post-mortem reviews as extended feedback*, [In:] *Advances in Learning Software Organizations*, pages 4-12 Springer 2003.
- [15] O. Salo, K. Kolehmainen, P. Kyllönen, J. Löthman, S. Salmijärvi, P. Abrahamsson, *Self-Adaptability of Agile Software Processes: A Case Study on Post-iteration Workshops*, [In:] *Extreme Programming and Agile Processes in Software Engineering*, volume 3092 of *Lecture Notes in Computer Science*, pages 184-193, Springer 2004.
- [16] B. Collier, T. DeMarco, P. Fearey, *A Defined Process For Project Postmortem Review*, *IEEE Softw.* **13**(4), 65-72 (1996).
- [17] F.F. Fajtak, *Kick-Off Workshops and Project Retrospectives*, [In:] *Professional Knowledge Management*, volume 3782 of *LNCS*, pages 76-81 Springer 2005.
- [18] T. Dingsøy, *Postmortem reviews: purpose and approaches in software engineering*, *Information and Software Technology* **47**(5), 293-303 (2005).
- [19] R. Scupin, *The KJ method: A technique for analyzing data derived from Japanese ethnology*, *Human organization* **56**(2), 233-237 (1997).
- [20] T. Schümmer, M. Mühlpfordt, J.M. Haake, *Computer Supported Reflection of Good Practice*, [In:] *Collaboration and Technology*, volume 6257 of *Lecture Notes in Computer Science*, pages 65-80, Springer 2010.
- [21] M. Margaritis, N. Avouris, G. Kahrmanis, *On Supporting Users' Reflection During Small Groups Synchronous Collaboration*, [In:] *Groupware: Design, Implementation, and Use*, volume 4154 of *LNCS*, pages 140-154, Springer 2006.
- [22] E. Bjarnason, A. Hess, R. Berntsson Svensson, B. Regnell, J. Doerr, *Reflecting on Evidence-Based Timelines*, *Software, IEEE* **31**(4), 37-43 (2014).



**Sylwia Kopczyńska**, graduated (*summa cum laude*) from Poznan University of Technology (PUT) with a degree in Computer Science. At present she is a PhD Candidate at PUT in the Institute of Computing Science. Her current research focuses on non-functional requirements engineering, software quality and measurement, and empirical software engineering methods. She has been participating in national and international projects, mainly as analyst, developer, tester in the domains of software development, insurance, transportation, travel, and pharmacy.