Principles and a Process for Successful Industry Cooperation – The Case of TUM and Munich Re

Maximilian Junker, Manfred Broy, Benedikt Hauptmann
Wolfgang Böhm, Henning Femmer, Sebastian Eder
Technische Universität München

Elmar Juergens
CQSE GmbH

Rainer Janßen
Rudolf Vaas
Munich Re

***Abstract*—Research transfer projects should be beneficial and inspiring for both, the academic as well as the industrial partners. If the setting is inadequate they can, however, also be a source of frustration and a waste of time and money for all parties. In the last decade, the Chair of Software and Systems Engineering at Technische Universität München (TUM) participated in a series of eight research transfer projects, conducted jointly with the re-insurance company Munich Re. The common theme of these projects has been quality of software development artefacts. This cooperation has been exceptionally productive for both sides. Results of this continuous success are, for example, a university spin-off, a considerable number of publications, as well as more systematic and improved methods in software engineering at Munich Re. A corner stone to the fruitful cooperation has been the model of how the university and practitioners have been working together. In this paper, we look at the cooperation from the retrospective and identify a number of basic principles that contributed to the success of the cooperation. Aditionally, we illustrate our research process and approach which helps to realize these principles.**

I. Introduction

Software Engineering deals with the question how to systematically create and maintain high-quality software. Although this question involves a considerable amount of basic research, Software Engineering is a research field with a strong relation to industrial practice of software development. It thus strives to solve problems that impair software creation and maintenance *in practice*. Hence, in order to produce relevant results, academia is bound to keep in touch with software industry. There are a number of ways to realize connections between industry and academia, such as conferences. Another form, which promises deep insights into industrial practice is the direct cooperation between a company and a research institute to work on problems together. From an academic perspective such a cooperation can be an opportunity to learn about problems occurring in practice, to gather real-life data, or to evaluate solutions proposed by research in an industrial environment. Benefits from an industrial perspective are to learn about new methods and technologies and to get insights and proposals for improvements from an unbiased, external observer but also to present the company to the scientific community and attract new highly qualified staff.

In the last decade the Chair of Software and Systems Engineering at Technische Universität München (TUM) has participated in a large number of research cooperations with industry partners. In many cases, these cooperations were very fruitful for both parties. In these cases, the cooperation often continued over several years, was able to improve the state-of-the-practice, resulted in a high motivation of the project team and contributed to doctoral theses and other research projects. In some cases, however, they turned out as a source of frustration and a waste of time and money as they did provide little value for both sides and consequently did not result in follow-up projects.

We believe that project success and disappointment were no coincidence. For this paper, we looked at one of our most successful research cooperations from the retrospective and extracted basic principles and a process that we believe played a major role in the success. Our partner in this cooperation is the re-insurance company Munich Re. Munich Re is one of the World's leading reinsurance companies with about 45,000 employees in reinsurance and primary insurance worldwide. For their insurance business, they develop a variety of individual software systems ranging from damage prediction, over pharmaceutical risk management to credit and company structure administration. Together, we cooperated in eight consecutive projects. The result of the last eight years from the academic perspective have been over 20 publications in international conferences and a university spin-off performing software quality assessment and improvement for Munich Re. From the perspective of Munich Re, the cooperation is perceived as a very productive undertaking, especially from a value for money perspective, compared to other consulting projects. The problems addressed over the years varied, but always in the context of arising challenges of ongoing software engineering projects. We were able to introduce several improvements in the software engineering methods of Munich Re.

Comparing this cooperations with others, we identified and extracted basic principles which we believe fostered the success. These principles relate to the mindset, the staffing, the project contents and the project organization. Over the years, we furthermore established a specific process to define new topics and execute the projects, which is in line with the principles and which equally contributed to the success of the cooperation.

II. Cooperation Principles

In the following, we present 12 principles that we think contributed to the success of our cooperation with Munich Re. We structure these principles along four classes which we took from classic project management: project content, staffing, organization, and mindset. It should be noted that the principles are not independent from each other. As a matter of fact, adhering to one principle may also have a positive impact on one or more other principles: For example, selecting the right project content (Principle 1) will increase the motivation of the project staff (Principle 9) and raise management attention (Principle 6).

A. Project Content

Principle 1: Select relevant problems: Selecting problems that are relevant for both, industry and academia, builds the foundation for a successful cooperation. The research on the problem should result in a solution with a direct and measurable benefit for the industrial partner. Selecting a relevant problem for industry will increase the chances of industry transfer and future cooperation. It makes it also easier to engage practitioners and thus create valuable feedback. Less obvious but equally important is selecting a problem relevant for academia: Choosing problems according to research relevance and interest, e.g. coordinated with dissertations or master theses ensures strong and sustained motivation on the academic side. Although the problems should be practically relevant, it is important not to keep the research project scope too specific, e.g. by dealing with a problem specific to only one internal project of the company. This can introduce unwanted risks, such as the loss of the only evaluation partner.

Principle 2: Select manageable-sized problems: Selecting problems with a manageable size is a simple risk-management method. Since research projects are inherently exploratory, they may always fail or not lead to the results desired initially. This may be due to unavailable/unreliable technology, unforeseen process problems or missing data. When the selected problems are small (e.g. in terms of effort to be spent), failure is less expensive. A follow up project with redefined goals can be set up based on the insight gained so far. In addition, small project iterations helps both sides to focus on quick-win problems with a high cost-benefit-ratio. Keeping the project duration short promotes choosing manageable problems. Our experience has shown that a scope of one year is a reasonable duration for a research cooperation. In our cooperation with Munich Re, we executed one project per year starting around February and finishing with a final presentation in December. Having small project scopes furthermore helped us to manage expectation and avoid unrealistic promises.

Principle 3: Choose concrete, non-invasive solutions: Solutions developed during the course of the projects have to be concrete and non-invasive. That means that, if applied, they will not require large changes in the process or tooling landscape. Discussing usage scenarios from early-on forces the academic team to adopt the perspective of practitioners and makes sure that the solutions are well suited to solve the problems at the side of the industry partner. It is a good practice to create a prototype for each solution, allowing to evaluate the idea from early on and gain concrete feedback, which in turn increases the chance of a successful research outcome. For example, many of our quality assurance concepts are analytical instead of constructive and most of our prototypes can be used in addition instead of as a replacement to the existing tools.

B. Staffing

Principle 4: Include industry staff in project team: Apart from the academic staff we always involved at least one employee from Munich Re into the project team. This person acted as an interface into the organization providing contact to internal software projects and gave valuable input with respect to the actual project content. It is important that the industry team member feels equally responsible for the project success.

Principle 5: Involve problem owners: Additionally to the contact person directly involved into the project, it is important to maintain direct contact to the problem owners at the industry partner. When, for example, we considered requirements quality in a project, we stayed in direct contact to several requirements engineers of Munich Re. This helped to ensure that the problems we dealt with were still relevant for Munich Re and the solutions we developed were adequate.

Principle 6: Involve management: A research project needs support through all hierarchy levels at the industry partner. Management level needs to provide budget, grant the right resources on the working level and always set the right priorities to make sure that the projects gets the right level of attention in the organization. This implies to include the project goals into the individual target agreement of the involved staff. This avoids the situation that tasks from all day business will always get higher priority. In case of the Munich Re cooperation we could rely on this support on all levels and the project goals were among the individual target agreements.

C. Organization

Principle 7: Maintain regular meetings: Regular meetings are held to review the project status and present preliminary results. Having regular meetings avoids loosing contact and helps to synchronize the expectations between the project partners. Early results can be discussed critically and, if necessary, changes to the originally planned solutions can be decided. In our case, we had a regular meeting every four to six weeks. In practice this interval was well-proven as it was large enough for significant progress to happen and still short enough to detect problems early.

Principle 8: Disseminate project results: Project results must be disseminated into the company as well as into the scientific community. By disseminating the project results, for example in the form of tools or presentations, the whole project team, academic and industry staff, can gain visibility inside the company. It is important that the dissemination activities focus on explaining why a project result is important and how it can help to improve day to day work in the company. Successful dissemination has several positive effects. First, a larger group is potentially interested in the project and its results. This again increases the acceptance of the research project in the company, creates potential new problem-owners or validation partners and thus advocates for follow-up projects. Second, a larger group can potentially profit from the research results, and, depending on the concrete results, use them to be more productive, deliver a better quality, etc. Publishing to the scientific community signals the relevance and level of innovation that the work represents. Furthermore, it is understood as a reward for the project team and keeps the motivation high. In our projects with Munich Re, we usually held several presentations during and after the project phase at internal events of Munich Re. Furthermore we strived to publish one or more scientific papers to international conferences. Usually these publications were a joint work between academic and Munich Re staff.

D. Mindset

Principle 9: Allow Creative Leeway: The involved people need to be given the necessary leeway for the project work and to create new ideas and solutions. On both sides, the involved staff needs to be granted enough resources to try something out. Especially if the initial solution idea does not work out, there must be room to change or abandon a topic and work in a different, more promising direction. Of course this needs to be decided together and a decision needs to be communicated as early as possible.

Principle 10: Trustful Interaction: Since the beginning of our cooperation, mutual trust in the professional abilities of both partners formed the basis for the cooperation. Munich Re appreciated the competence of our research group to achieve state-of-the-art research results as well as our confidential treatment of the entrusted data. In return, we appreciated Munich Re's long term experience in software development. This led them to give us broad access to almost any kind of data available, such as, the source code of their software system or business specifications and test cases which was necessary to successfully realize the project (see Principle 1). Furthermore, it enabled us to focus on problems which are interesting from a scientific perspective as well to work on Munich Re's concrete problems.

Principle 11: Openness for Criticism: In order to assess the practical value of a novel solution, the involved company needs to provide an open minded-environment supporting constructive criticism. This is the precondition for objective evaluation of research results. In our case, a core motivation Munich Re was to learn about their problems and ways to solve them. This led them, for example, to appreciate analytic results even if they revealed issues with current artefacts and processes.

Principle 12: Appreciate Mutual Interests: Both partners need to understand and respect the motivation and the interests that each partner connects with the cooperation. For example, Munich Re supported our need to answer questions related to doctoral theses and to publish research results. On the other hand, we appreciated their interest in achieving practically usable solutions but also, for example, to get into contact with students and to arouse their interest in practically important topics.

III. Cooperation Process

In our cooperation with Munich Re we made good experience with the following steps performed to plan and execute the projects.

A. Identify Problems

The first step in each cooperation was to identify problems which are both, relevant for Munich Re and promising in terms of research output. Munich Re as well as the academic researcher team performed the search for candidate problems together. Due to experiences gathered from projects with Munich Re or other, similar, companies, we were usually able to propose a number of problems that seemed relevant from the perspective of Munich Re. Often topics for a project emerged during the predecessor project.

B. Propose and Evaluate Possible Solutions

For each problem candidate we proposed possible solution approaches together with the practitioners. Therefore, we usually studied available literature on the topic and reviewed the current state of the art. With this approach we could gain confidence that there are no fundamental obstacles associated with solving the problem. Often we created small prototypes to assess the feasibility of a proposed solution. For example, in case of the topic requirements traceability recovery we proposed to use natural language processing techniques in order to uncover traces between artefacts. The researcher team then created a minimal prototype that was capable to relate use case documents with manual test case descriptions.

C. Create Research Agenda

Based on these preliminary steps practitioners and researchers decided together on the research agenda for the next project phase. In each of the projects that we conducted so far, the project phase has always been one year. From the list of the problem candidates and solution ideas we typically distilled three to four work packages. In a next step we created a document where we described for each work package the problem, its context, a proposed solution approach, the concrete steps that we anticipate, the deliverables that will be created during the project phase as well as estimated effort in person months. In the majority of cases the final deliverables included a tool or a tool extension. Often, the tool implemented some kind of analysis, such as the aforementioned traceability recovery analysis.

D. Realize Solution

Every work package was structured as a mini project. The steps for each work package therefore usually included a preparation step, the actual creation of the solution as well as a scientific validation of the solution in form of a study.

1) Preparation: For every work-package, we talked to the relevant stakeholders, such as team members of the software projects we planned to use as pilots, or screened relevant artefacts (e.g. documents, source code). This step is important to ensure that the proposed solution can be applied.

2) Create Solution: After the preparation we created the actual solution. During this phase we held a number of regular meetings with the whole project team. Additionally, as soon as first prototypes (in case of tools) or first analysis results were available we organized separate meetings with the problem owner and discussed the results.

3) Validate Solution: The final step usually was to validate the solution. This does not only involve ensuring that the solution works technically, but primarily to study if it indeed solves the initial problem. Often, we initiated an empirical study where we applied our solution to concrete artefacts of Munich Re. We thus investigate the feasibility of the solution. Furthermore we usually performed formal interviews to assess the acceptance of the solution in practice.

E. Disseminate Project Results

We disseminated the project results in two ways, by presentations and publications as well by providing tools.

1) Presentations and Publications: We presented the project results, both internally at Munich Re as well as externally, for example at research conferences. Regarding the internal presentations there is usually one final presentation attended by staff from all hierarchy levels up to the CIO. Additionally we regularly presented the results in internal special-interest communities (such as a Requirements Engineering community). Finally, selected topics were usually published at scientific software engineering conferences and workshops.

2) Provide Tools: As a second way of dissemination we often provided tools. Our goal always was to bring these tools into the productive use after the project. When our solution involved automated analyses of certain artefacts, these analyses were installed at Munich Re and configured such that they run every day and provide analysis results to the software project team members.

IV. Two Examples

In the following we illustrate some principles we presented so far with two examples from the Munich Re/TUM cooperation. We will first briefly introduce two topics on which we were working on with Munich Re in the last years.

A. Case "Test Gap Analysis"

The initial goal in this case was to support test engineers of Munich Re in aligning test efforts with maintenance efforts. The specific challenge in this case is that development and testing are in general performed by different contractors. As there was a considerable amount of manual tests, the testing efforts should concentrate on the parts of the application that have actually changed. We suggested to approach this problem by automatically identifying areas that have changed compared to the last release, but have not yet been tested.

We worked on aspects of this problem in three consecutive projects, resulting in an analysis tool that is now deployed for several productive applications at Munich Re as well as the publication of an empirical study [1]. The work is furthermore part of a doctoral dissertation.

B. Case "Maintainable Tests"

In our second case, we addressed the challenge of reducing maintenance costs of system test suites. In a study performed with Munich Re in 2011 we found that system tests contain many clones which blow up test artifacts and make them complex to understand and maintain [3]. Together with Munich Re, we decided to approach the problem both analytically and constructively. In the analytical branch we devised analyses looking for clones and other quality defects automatically. In several projects we developed and evaluated "test-smells" for various test artefacts [2]. These smells are currently at the edge of being broadly introduced within Munich Re.

In order to address this issue constructively, we designed a new test definition tool which provides adequate reuse mechanisms (which were lacking in the current tooling), and actively supports test engineers in avoiding clones. However, even though our first experiments have been promising, we decided to stop working on this topic after two years. The main reason for this decision was that the efforts for developing a production strength authoring tool would have bound resources for years and thus prevented us from working on other topics together with Munich Re.

C. Discussion

The cooperation in the first case (*test gap analysis*) has been beneficial for Munich Re as well as for us researchers. It illustrates how we approached a problem originating from Munich Re and in three consecutive projects incrementally created a solution that directly helps the company, at the same time advanced the state of the art (Principles 1 and 2). From the beginning we were able to test our ideas using real-live data and discuss results with testing staff (Principle 5). After proving that our analysis tool does not interfere with the application's functionality we could even install it into the testing environment of Munich Re (Principle 10). Information about possible testing gaps were greatly appreciated by Munich Re (Principle 11). The test gap analysis is currently deployed for several Munich Re projects and is supervised by Munich Re quality engineers (Principle 3, 8).

The second case is more ambivalent. For the analytical approach, we made a similar experience as in the first example. The constructive approach, however, was stopped after two years and did not transition into productive use. The reason for this is not that the problem was less relevant or because the solution was not adequate. Instead, it is an example for not violating Principles 2 and 3. Developing a production ready tool is no longer a small-size venture. Our solution would replace existing tooling and therefore had to provide a considerable set of basic features immediately and reliably. Furthermore, introducing the tool would have been much more invasive than introducing an analysis: Although it sounds promising, it interferes with established tools and processes.

V. Summary

Cooperation with industry is a promising way for a research organization to identify problems that are pressing in industrial practice, to gather real-life data about software development and to validate new methods or tools. For industry it is an opportunity to improve software engineering methods and to get in contact with researchers and students. However, cooperations between industry and academia sometimes fail. In this paper we discuss 12 principles and a process that we believe contribute to successful cooperations. We develop both from a retrospective of an eight-year long successful cooperation with Munich Re and the Technische Universität München.

References

[1] Sebastian Eder, Benedikt Hauptmann, Maximilian Junker, Elmar Juergens, Rudolf Vaas, and Karl-Heinz Prommer. Did we test our changes? assessing alignment between tests and development in practice. In *AST*, 2013.

[2] Benedikt Hauptmann, Maximilian Junker, Sebastian Eder, Lars Heinemann, Rudolf Vaas, and Peter Braun. Hunting for smells in natural language tests. In *ICSE*, 2013.

[3] Benedikt Hauptmann, Maximilian Junker, Sebastian Eder, Elmar Juergens, and Rudolf Vaas. Can clone detection support test comprehension? In *ICPC*, 2012.