Facts and Fallacies of Reuse in Practice

Veronika Bauer
Technische Universität München, Germany
bauerv@in.tum.de

Abstract—Despite the positive effects of reuse claimed in a significant amount of research, anecdotal evidence indicates that industry is not yet experiencing the expected benefit. This dissertation proposal aims to investigate these indicators and therefore addresses reuse from an industrial point of view. As a first step, it empirically assesses the general state of reuse in practice. This is achieved via a large-scale online questionnaire distributed to multiple companies. Complementing the questionnaire, extensive interviews are being scheduled with developers and project managers of the respective companies. The goal is to interview approximately 10 employees per company. Three companies have already committed to the interview phase and contact with seven further companies is currently being established. In a second step, the findings of the study will be used to extract the context and type of reuse, as well as success factors and hindrances. This information forms the basis for an analytical assessment model for internal code reuse, which is developed in a third step. It will capture a range of different aspects of reuse in practice and will be combined with a process to evaluate the adequacy of reuse. The planned result is a larger assessment framework for evaluating the reuse (management) process within a project as well as a multi-project context. As a result, guidelines for code organization should be developed and tested for their effects in improving reuse in one or more of the industrial partners projects.

Keywords—software reuse, internal, industrial, maintenance, software quality

I. INTRODUCTION

Reuse of development artefacts has become a standard strategy in large-scale software development to deliver feature-rich software in a cost-efficient and timely manner. Libraries and frameworks represent a great opportunity for productivity improvement and also can improve overall code quality [1], [2].

However, anecdotal evidence suggests a heterogenous reuse experience in industrial practice: Whilst some software companies have established successful reuse strategies, many others do not yet experience the full potential promised by reuse. A survey by Mohagheghi and Conradi supports this indication for the time between 1994 and 2005 [3].

An important question is therefore: How can the benefit of reuse be increased in industrial practice?

A significant amount of research has been done in the field of software reuse for a long time, focussing on aspects of the retrieval of reusable entities via code search engines and source code mining, assessing the reusability of artefacts, programmer support with e.g., code recommenders [4], quantification of library reuse via reuse rates in open source systems [5]–[7]. Also organisational factors have been considered [8]. Recently, the focus has shifted to identifying maintenance consequences and potential risks introduced by library reuse [9]–[12]. Consequently, the quality assessment of reuse aspects is emerging as research direction [13].

In this dissertation, the goal is to propose an activity-based assessment framework (cf. [11], [14]) for code reuse. As a foundation, factors influencing reuse in practice will be collected within an explorative empirical study, which will assess the state of reuse in industrial development practice. Furthermore, the outcomes of the study will serve as the basis for hypothesis generation on success factors and hindrances of beneficial reuse. The hypotheses will then be subject to technical case studies on industrial systems with the goal of validation or rejection. The assessment framework will integrate the insights of the studies and provide measures to quantify the influence factors. In addition, a process will be provided to support assessing code reuse in practice. Aspects of and strategies for cross-project reuse will be emphasized, as most of the existing work focusses on assessing reuse at a project scope.

The steps of this dissertation proposal are:

- Understand how reuse occurs in industrial practice, especially which environments and ecosystems foster or hinder reuse.
- From the information gained in the first goal, build hypotheses and extract success factors as well as hindrances of reuse.
- Develop and validate an assessment framework for reuse in industrial context.

Outline: Section II gives an overview on related work. Section III details the contributions achieved so far and describes the work planned in the next steps. Section IV lines out some of the risks of the proposed methods.

II. RELATED WORK

A. Reuse in industry

Some work has been done in assessing the state of the practice of reuse: In his PhD dissertation [15], Stuetzle empirically assesses reasons for and against reuse in an industrial context. He addresses the economical aspects of reuse and provides a cost model for estimating the effort...
involved when building for reuse as well as consuming reusable artefacts. My dissertation proposal differs from Stuetzle’s dissertation in that it aims to gain insights into the actual reuse situations currently prevalent in industry. Furthermore, his conclusions on economical feasibility could be built upon in the envisioned assessment model.

A small example of a study on reuse in industry has been provided in the diploma thesis of Ioana Kalaydjieva [16]. Kalaydjieva studied the code reuse behavior prevalent in the Softlab GmbH and provided steps to improve the company’s specific reuse strategies. My dissertation proposal envisions an empirical evaluation of reuse behaviour at a larger scale: on the one hand, the focus is going to be wider than just code reuse, including other development artefacts. On the other hand, the number of companies participating in the studies should be at least 10, from different business context.

B. Organizing for internal reuse

The main focus of this proposal is on internal reuse, investigating how companies manage their artefacts to enable reuse within and across projects.

Basili et al. propose an “infrastructure”, called the “experience factory” [8] to organize knowledge and artefacts obtained throughout the development process of software systems. They postulate a logical separation between development and knowledge/analysis organizations, which follows different incentives and goals. This work is interesting in the context of this proposal, as it sketches out an organizational model for enabling company-wide internal reuse. It, however, demands a significant degree of formalization and modelling capabilities, which might not be prevalent in the average software company. In the studies conducted within my dissertation project, the internal organization of companies will be assessed to see whether structures and strategies matching the experience factory are present.

C. Third-party reuse

A significant amount of work has addressed extent, support, and assessment of third-party reuse in Open Source projects. In my dissertation project, the focus will be on assessing the use of third-party libraries and their impact, rather than the support to employ them. However, some work on the extent or the functionality of library reuse will be used as comparative to the results drawn in industrial projects.

Assess reuse In a recent study, Heinemann et al. quantified the reuse via APIs in Java open source projects and found that for almost half of the projects the amount of reused code exceeded the amount of newly developed code [5]. Furthermore, they classified the different types of functionality which was typically reused.

Lämmel et al. [7] analyzed API usage in 1,476 open source Java projects. They determined the API usage footprint of the projects, determined by the number of included libraries and the number of (distinct) API methods that are called from the projects’ code.

Klatt et al. [12] suggested an approach to identify the impact of evolving third-party components on long-living software systems. They use a white-box impact analysis which requires access to the third-party source code and combined it with data from bug trackers and quality analyses on the third-party code.

Kotonya and Hutchinson [9] suggested an approach that helps developers understanding the impact of change in commercial off-the-shelf (COTS) software components employed in a project.

Raemaekers et al. [17] proposed an approach to automatically assess the risk imposed by third-party library usage in software projects. They measure the usage frequency of third-party libraries in a corpus of open source and commercial software systems. The risk assessment is based on the assumption that an uncommon, i.e., infrequently used, libraries expose a higher risk compared to a library that is frequently employed by software projects.

III. CONTRIBUTIONS

This proposal contains the planned contributions relating to reuse implementation in an industrial setting. Figure 1 highlights the three conceptual steps: first, a survey assesses reuse in practice on a very broad level, comprising constructive and analytic aspects. Second, success factors, hindrances, and potential risks of reuse will be extracted. Importance will be given to find measures to quantify them. The extraction will be based on the survey as well as on static analyses of the available code bases. Third, we address the often-neglected step of assessing reuse within a project by providing an assessment model and process for industrial application.

A. Accomplished contributions

My work so far covers aspects of reuse assessment with respect to third-party library usage. It will be extended
and integrated in the assessment model envisioned in this proposal.

Quantification of third-party library reuse: In [13], we proposed an approach to determine the degree of dependence between a software project and its third-party libraries in order to support decision making in various use cases during software maintenance. Our approach provides information on characteristics regarding third-party library reuse which are non-trivial to manually determine for large systems, e.g., the quantification of library usage, scatteredness and entangledness of libraries and system.

We provide open source tool support for automated analysis and visualization of the results for single software projects\(^\text{1}\). The approach allows for substantiated decisions in software maintenance scenarios, such as API migration and evolution. The findings of an evaluation on open source software systems indicate that the results of our analysis provide meaningful answers in the identified usage scenarios.

This approach is now integrated in our structured assessment model presented in the next section.

Structured approach to assess third-party library usage: Based on industry needs, in [11] we proposed a structured approach for the systematic assessment of third-party library usage in software projects. The approach is supported by a comprehensive assessment model relating key characteristics of software library usage to development activities. The model defines how different aspects of library usage influence the activities and, thus, allows to assess if and to what extent the usage of third-party libraries impacts the development activities of a given project. Furthermore, we provided guidance for executing the assessment in practice, including tool support for a pre-selection of important libraries and multiple automated static code analyses. We evaluated the approach with a case study in a consulting context. In collaboration with a consultant, we assessed an industrial software system of 3.5 MLOC including about 90 external libraries. Our approach was perceived as helpful and a significant improvement over the methods employed so far for assessing third-party library usage.

B. Current Work

Currently, I am working on the empirical study to collect influence factors on reuse with the goal to extend the assessment approach on third-party library usage with concepts to model and assess internal code reuse.

Study of reuse in industrial context: To gain a deeper understanding of reuse in industrial praxis, an empirical study is currently being designed and piloted. It consists of two parts: qualitative interviews and a quantitative online questionnaire. Both parts comprise various aspects of reuse, which have been collected during a preliminary literature study, such as motivation for reuse, organizational factors, technical realization and infrastructure, tool support, reuse processes, problems caused by reuse, quality assurance of reuse, legal aspects, and maintenance with reuse.

The first round of interviews is currently being conducted with 10 developers and project managers of an international software development company. Each interview is scheduled to last 60 minutes and, apart from project and developer information, consists of open questions only. Per interview, two researchers are present, one leading the interview, one taking notes and summarizing. The topics mentioned above are addressed from two points of view: first, we collect information on how the company is currently addressing the different aspects of reuse. Second, to collect hints for further research directions, we ask for opinions on success factors and hindrances, as well as fictional wishes (“Independent of its realizability, what would you wish for to make reuse beneficial in your work?”). After each interview, the researchers extract the obtained statements and, after discussing and indexing them, cluster them into topics to prepare the hypothesis generation (see Figure 2). Further interview rounds will follow with additional industrial partners from different domains.

The goal of the online questionnaire is to collect quantitative data on the same aspects of reuse as the interviews. It will be distributed over a large network of software companies. The questionnaire is currently in a pilot-phase with an industrial partner and will be refined after receiving their feedback. Also insights from the interviews will be taken into account. Subsequently, we will distribute the questionnaire to developers and project managers over a large network of software development companies. With interested respondents complementary interviews will be scheduled to clarify points raised in the survey and to better understand the respective artefact management strategies of the different software companies.

C. Planned Work

The next steps following the survey will be the extraction of success factors, hindrances, and measures, as well as the development of the assessment model. The model will

\(^{1}\)The analyses are integrated in the quality framework ConQAT, available from www.conqat.org
include the dependencies between software reuse characteristics and software quality as well as development efficiency and maintenance activities. Inherently, analyses will be run on the available industrial systems to validate hypotheses and measures for influence factors. Furthermore, it is envisioned to evaluate reuse specific metrics, such as reuse rate or entanglement over time. This would enable to link significant reuse changes found in the repository with organizational influence factors. Also, cross-project reuse behaviour will be addressed. Iteratively, the assessment model will be refined according to the outcomes of the different analyses. Furthermore, guidelines and recommendations for code-base management will be derived. To complete this work, an empirical study is envisioned to validate the framework management will be derived. To complete this work, an empirical study is envisioned to validate the framework

Furthermore, the assessment model will be refined according to the outcomes of the different analyses. Additionally, guidelines and recommendations for code-base management will be derived. To complete this work, an empirical study is envisioned to validate the framework and guidelines in practice with technical case studies. It is planned to introduce them in a specific company, to monitor the process and to measure quality indicators such as clone coverage and reuse rate to verify if our hypotheses hold.

IV. THREATS

This proposal relies heavily on empirical methods and is therefore dependent on participant response in the survey. Two strategies are being followed for recruiting participants: direct contact to current or former project partners and dissemination via a large business network, located at Technische Universität München. The supporting interviews might be biased, as participants might opt in only if they have strong opinions (which, however, might also balance between pro or contra reuse statements). In order to achieve a minimum of generalizability, we attempt to conduct our studies with as many industrial partners as possible. Currently, three software companies have agreed to participate, with seven further candidates pending.

V. CONCLUSION

This work investigates reuse of source code primarily in the context of industrial software systems with the goal of understanding chances and risks of reuse, best practices and hindrances, as well as the circumstances required for beneficial reuse to occur. Building on an empirical evaluation of the state of the practice, it will provide an assessment model capturing relevant aspects of reuse in practice, and their influence on development and maintenance. In addition, it will develop a process applicable in industry, with the goal to increase the benefit of reuse.

REFERENCES