

Binish Tanveer

Utilizing Change Impact Analysis for Improving Effort Estimation in Agile Software Development



Editor-in-Chief: Prof. Dr. Dieter Rombach
Editorial Board: Prof. Dr. Peter Liggesmeyer
Prof. Dr. Frank Bomarius

FRAUNHOFER VERLAG

PhD Theses in Experimental Software Engineering

Volume 66

Editor-in-Chief: Prof. Dr. Dieter Rombach

Editorial Board: Prof. Dr. Frank Bomarius
Prof. Dr. Peter Liggesmeyer
Prof. Dr. Dieter Rombach

Binish Tanveer

Utilizing Change Impact Analysis for Improving Effort Estimation in Agile Software Development

Fraunhofer Verlag

Zugl.: Kaiserslautern, TU, Diss., 2019

Printing:
Mediendienstleistungen des
Fraunhofer-Informationszentrum Raum und Bau IRB, Stuttgart

Printed on acid-free and chlorine-free bleached paper.

All rights reserved; no part of this publication may be translated, reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. The quotation of those designations in whatever way does not imply the conclusion that the use of those designations is legal without the consent of the owner of the trademark.

© by **Fraunhofer Verlag**, 2020
ISBN (Print): 978-3-8396-1568-3
Fraunhofer-Informationszentrum Raum und Bau IRB
Postfach 800469, 70504 Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon +49 711 970-2500
Telefax +49 711 970-2508
E-Mail verlag@fraunhofer.de
URL <http://verlag.fraunhofer.de>

Utilizing Change Impact Analysis for Improving Effort Estimation in Agile Software Development

Thesis approved by
the Department of Computer Science
Technische Universität Kaiserslautern
for the award of the Doctoral Degree
Doctor of Engineering (Dr.-Ing.)

to

Binish Tanveer, M.Sc.

Date of Defense: 10.09.2019

Dean: Prof. Dr. Stefan Deßloch

Reviewer: Prof. Dr. Dr. h.c. Dieter Rombach

Reviewer: Prof. Dr. Kai Petersen

Abstract

Effort estimation is used for planning and managing all phases of software development. Traditional estimation methods rely on having a mostly “complete” and “fixed” specification of a system. In agile software development, cost and effort estimation is challenging, as the requirements cannot be specified entirely upfront and are evolved as development progress and users interact with the product. It is imperative that the scope of a requirement is sufficient for completing it within a time-boxed iteration called sprint. Thus, estimation is equally essential for sprint planning.

Current practice in this context relies heavily on human judgment which has several limitations. It is based on limited information (subjective opinion), hampered by wishful thinking and prone to human judgment bias (individual and group effects). Estimates produced by these methods are often inaccurate as they are made with limited implicit knowledge leading to underestimation. Experts recognize the need and importance of more objective information that includes historical data from previous iterations. Moreover, current estimation methods do not objectively consider the potential impact of a change on existing software and custom factors that contribute to the effort overhead. In conclusion, first, there is a need for systematically storing historical data, including estimates together with the implemented change and contextual information, in an experience base for learning and reuse purposes. Second, there is a need for systematic tool support for analyzing the potential impact and required effort of changes. Thus, considerable improvement potential exists concerning systematic effort estimation in this environment and marks the contribution of this research to the body of knowledge.

This thesis introduces HyEEASe - a hybrid, lightweight and systematic method for estimating the effort in the context of iterative, incremental software development. It utilizes change impact analysis information for supporting estimation by human judgment. Additionally, a solely data-driven estimation model based on Gradient Boosted Trees (GBT) is also developed. HyEEASe was evaluated using two case studies and a controlled experiment. The GBT estimation model was evaluated against Agile COCOMO II. The results indicate that the proposed hybrid method is useful and it produces more effective estimates than purely expert-based or purely model-based estimates. Furthermore, it provides more useful impact information and enables learning of impact information. The performance of the GBT model outperformed Agile COCOMO II regarding estimation effectiveness.

Acknowledgements

Firstly, I would like to thank my principal advisor, Prof. Dr. Dr. h.c. Dieter Rombach from the Technische Universität Kaiserslautern, Germany, for giving me this opportunity to conduct empirical research in a real-world setting. Then I am indeed thankful for his continuous support and guidance throughout the thesis that certainly improved the quality of the research. I want to thank Dr. Jens Heidrich whose thorough involvement and guidance helped bring ideas to practice. From general discussions till the writing of the thesis, he had been there supporting me always. I am also thankful to Dr. Kai Petersen from Blekinge Institute of Technology, Sweden, for his rigorous review and feedback that helped greatly in shaping the thesis.

At the university group and Fraunhofer IESE Kaiserslautern, I am thankful to the colleagues who participated in the extensive discussions. I am indeed grateful to Dr. Adam Trendowicz, who had been the main driver in guiding me in the right direction to achieve the goal. He had been there since the inception of the thesis idea. He also greatly helped in introducing and evaluating the project in industrial contexts. I also want to thank Dr. Liliana Guzmán, whose support in various activities of the thesis, including empirical validations, was beneficial. Her support in designing empirical studies is commendable.

I would also like to mention the German Ministry of Education and Research (BMBF) as part of this research was supported by projects including HyEEASe, grant number 01IS12053 and Abakus, grant number 01IS15050G. I am thankful to all industrial and research partners in these projects who gave me a platform to develop and evaluate this work in an industrial context. I also wanted to thank my co-authors on the publications that resulted from this thesis.

I want to thank my parents, brothers, and friends for being big support throughout. Last but not least, to Nauman and kids for their patience, support, and appreciation that kept me going forward in this long journey.

Table of Contents

1	Overview	1
1.1	Introduction.....	1
1.2	Motivation.....	1
1.3	Research scope and approach	2
1.4	Research problem.....	4
1.5	Goals and hypotheses	5
1.5.1	Goals	5
1.5.2	Hypotheses	6
1.6	Evaluation strategy.....	6
1.7	Background	7
1.7.1	Effort estimation methods.....	8
1.7.2	Change impact analysis	9
1.8	Research contribution.....	10
1.9	Thesis structure	11
2	State-of-the-practice	13
2.1	Introduction.....	13
2.2	Related literature	13
2.2.1	Estimation methods issues	14
2.3	Case study	15
2.3.1	Goal.....	15
2.3.2	Sample and population	16
2.3.3	Design	16
2.3.4	Data collection	17
2.3.5	Data analysis	18
2.3.6	Threats to validity	19
2.3.7	Results - estimation method.....	19
2.3.8	Results - estimation method issues.....	20
2.4	Survey	22
2.4.1	Goal.....	22
2.4.2	Sample and population	22
2.4.3	Design	23

	2.4.4 Data collection	24
	2.4.5 Data analysis	25
	2.4.6 Threats to validity	25
	2.4.7 Results - estimation method.....	25
	2.4.8 Results - estimation method issues.....	26
2.5	Estimation method issues - literature, case study, and survey.....	27
2.6	Requirements of estimation method.....	28
	2.6.1 Mapping of requirements to problems and goals .	28
2.7	Summary	31
3	State-of-the-art.....	33
3.1	Introduction.....	33
3.2	Review of effort estimation in agile software development	33
	3.2.1 Background.....	33
	3.2.2 Review design and process	34
	3.2.3 Results	35
3.3	Evaluation of existing effort estimation methods.....	38
3.4	Review of change impact analysis	41
	3.4.1 Background.....	41
	3.4.2 Review design and process	42
	3.4.3 Results	43
3.5	Summary	47
4	HyEEASe - Method for effort estimation in agile software development	49
4.1	Introduction.....	49
4.2	Conceptual framework	50
4.3	Selected impact analysis and effort estimation techniques	53
4.4	Overview of the design	56
4.5	Impact analysis model development - Process A.....	56
4.6	Estimation model development - Process B.....	61
4.7	Detailed work flow - Process C (Perform estimation)	63
4.8	Estimating a user story using HyEEASe - an example	66
4.9	Assumptions /applicability of HyEEASe.....	71
4.10	Summary	71

5	Empirical evaluation	73
5.1	Introduction.....	73
5.2	Evaluation strategies	73
5.3	Case study at SAP SE.....	74
5.3.1	Example scenario - context and selected techniques.....	74
5.3.2	HyEEASe mock-up.....	76
5.3.3	Evaluation approach	79
5.3.4	Evaluation criteria	80
5.3.5	Evaluation design - sample and population	80
5.3.6	Execution	80
5.3.7	Data collection and analysis.....	81
5.3.8	Results and interpretation	81
5.3.9	Threats to validity	82
5.3.10	Conclusion.....	85
5.4	Case study at Insiders Technologies.....	85
5.4.1	Refinements of HyEEASe	86
5.4.2	Evaluation object.....	86
5.4.3	Evaluation approach	86
5.4.4	Evaluation criteria	87
5.4.5	Evaluation design - sample and population	88
5.4.6	Organizational context.....	89
5.4.7	Execution and data collection	89
5.4.8	Data analysis	90
5.4.9	Results and interpretation	90
5.4.10	Threats to validity	94
5.4.11	Conclusion.....	96
5.5	Data-based evaluation	96
5.5.1	Evaluation goals and hypotheses.....	97
5.5.2	Evaluation criteria	97
5.5.3	Agile COCOMO II.....	97
5.5.4	Set up of Agile COCOMO II.....	98
5.5.5	Results and comparison.....	98
5.5.6	Conclusion.....	99
5.6	Controlled experiment with students	100
5.6.1	Motivation and context selection	100
5.6.2	Experiment design	101
5.6.3	Execution	104

5.6.4	Threats to validity	106
5.6.5	Analysis and interpretation.....	108
5.6.6	Conclusions	113
5.7	Summary	115
6	Conclusions and future work	119
7	References	123
A	Appendix A.....	133
B	Appendix B	151
C	Appendix C.....	161
D	Appendix D.....	177
E	Appendix E	187
E.1	COCOMO II Reuse model	187
E.2	Set up of Agile COCOMO II - scale factors and cost drivers	188
F	Appendix F	193
	Curriculum Vitae.....	227

List of Figures

1.1	Research approach	3
1.2	Mapping of problems to goals and hypotheses.....	7
1.3	Empirical validation strategies	8
4.1	Framework for integrating impact analysis with estimation	50
4.2	Integration of IA with EE techniques.....	51
4.3	High-level design overview	56
4.4	Output of A.1 and A.2 - Example illustration	59
4.5	Output of A.3 - Example illustration.....	60
4.6	GBT model - Tree 1 view	64
4.7	Detailed work flow of Process C	65
4.8	Output of C.1 Impact set identification and C.2 Apply impact analysis.....	67
4.9	C.3 Revise impact analysis results - 1	69
4.10	C.3 Revise impact analysis results - 2.....	70
5.1	Empirical evaluation strategies	74
5.2	Mock-up workflow	76
5.3	Mock-up user interface - initial screen	77
5.4	Mock-up user interface - dependency graph view	78
5.5	Mock-up user interface - impact view	79
5.6	Comparison of estimation accuracy.....	92
5.7	Experiment execution	106
5.8	Testing the H3 - Usefulness (understandability) of estimation methods across both the groups.....	109
5.9	Testing the H4 - Learnability of estimation methods across both the groups	111
5.10	Students in both the groups providing rationale to justify their effort estimate	112
5.11	Perception of both groups on H3 and H4.....	112
5.12	Perception about H3 (understandability, completeness) and H4 (learnability) across and with in the groups	113

5.13 Student feedback regarding understandability of the method	114
--	-----

List of Tables

2.1	Sample	17
2.2	Factor definition.....	23
2.3	Factors comparison.....	24
2.4	Identified issues of estimation methods.....	27
2.5	Requirements regarding estimation method.....	29
2.6	Mapping of requirements to problems and goals.....	30
3.1	Papers searched through snowballing	36
3.2	Estimation method evaluation criteria.....	39
3.3	Estimation methods evaluation	40
3.4	Papers searched through snowballing	43
4.1	Code-based impact factors and their metrics	54
4.2	GBT estimation model parameters input - an example.....	70
5.1	Results of evaluated quality aspects of HyEEASe method and the mock-up (median values)	83
5.2	Positive feedback given by practitioners	83
5.3	Practitioners' suggestions and authors' response	84
5.4	Comparison of estimation accuracy.....	91
5.5	Results of the evaluated quality aspects of the prototype.....	93
5.6	Positive feedback given by practitioners	94
5.7	Practitioners' suggestions and authors' response	95
5.8	Evaluated research goal and hypotheses	97
5.9	Estimated effort for the tasks using Agile COCOMO II.....	99
5.10	Comparison of estimation accuracy - H1	99
5.11	Comparison of results of hypotheses across case studies	101
5.12	Mapping of tasks to evaluated hypotheses.....	103
5.13	Experiment design in detail	104
5.14	Experiment group details	105
5.15	Evaluation summary	115
5.16	Estimation methods evaluation	117
E.1	Agile COCOMO II Scale factors ratings	188

E.2	Agile COCOMO II Cost drivers ratings.....	188
-----	---	-----

1 Overview

1.1 Introduction

This chapter introduces the research done in the context of this thesis. It starts with a brief introduction of effort estimation in an agile development context and describes the associated problems and challenges. It details the research goals and hypotheses and the strategy adopted to address them. Furthermore, it highlights the research contributions of the thesis. It concludes with an overview of the contents and a presentation of the structure of the thesis.

1.2 Motivation

Unlike traditional software development approaches, agile embraces change. The requirements typically cannot be specified entirely upfront and are developed as the project progresses. The resulting dynamism of requirements makes estimating effort accurately a challenge. The software development and delivery in this context are structured in short time-boxed iterations often referred to as sprints. The scope of a requirement must be sufficient for completing it within a sprint. Thus, estimation is equally important for sprint planning. Each sprint involves planning, development, integration, testing and delivery [87]. Incorporating a single change could entail a significant impact across the various individual software components/modules, which makes it an important aspect to consider while planning and estimating effort.

Currently, effort estimation in this context relies heavily on human judgment. Typically, a cross-functional team of experts estimates by building consensus on how much effort a particular change will entail. This approach is labor-intensive, and due to the use of limited information, it has limited prediction accuracy.

The systematic analysis of the impact of a change can be used to estimating its costs and facilitate planning [73]. Change impact analysis techniques identify software life cycle objects that are likely to be affected by a given change request. Bohner et al. [35] have also suggested the use of impact analysis in supporting practitioners to precisely determine the effort and cost estimates for a software change. None of the existing estimation methods (in traditional or agile development) so far have considered the quantification of the impact that a change has on existing software. Therefore, an improvement potential exists concerning systematic effort estimation in this environment.

1.3 Research scope and approach

The research reported in the thesis is in the area of expert-based software effort estimation in the particular context of agile software development. With the focus on expert-based effort estimation methods, this thesis characterizes the explicit information required by the experts as they perform the task of estimation.

In the state-of-the-art and practice analysis as part of the foundation work for this thesis, it was recognized that while many of the effort drivers are either not relevant to the agile context or are already considered when estimating effort. However, the impact of a change on an existing system is not objectively quantified. Recognizing that the impact of a change on an existing system is a significant information source in effort estimation, this thesis makes use of the research done in the area of change impact analysis. The premise is that if the information about the impact of a change is objectively provided to the experts while performing estimation, it can support them to make informed decisions. It is also hypothesized that it will improve estimation accuracy and reduce bias.

This thesis considers various industrial application domains mainly management information systems. The geographical scope of this thesis comprises software companies in Germany like SAP SE [7] and Insiders Technologies GmbH [3].

This research is an attempt to identify the limitations of expert-based estimation methods in practice, analyze the solutions proposed in the published literature to improve these methods, and identify research gaps. It further proposes and evaluates in practice, a systematic effort estimation method that addresses the identified practical challenges and knowledge gaps.

The research approach pursued in this thesis is shown in Figure 1.1 and consists of the following steps:

1. Problem identification:
The current state-of-the-practice of effort estimation was analyzed concerning industrial requirements, currently applied estimation methods as well as problems associated with the methods and their output, using industrial case study and survey with questionnaire and interviews as well as the issues as reported in the related literature.
2. Solution idea:
Through state-of-the-art analysis, existing estimation methods in the context of agile software development were reviewed and analyzed for their demonstrated ability to solve issues related to expert-based estimation in practice. Furthermore, as it is claimed that change impact analysis can be used for estimating the effort

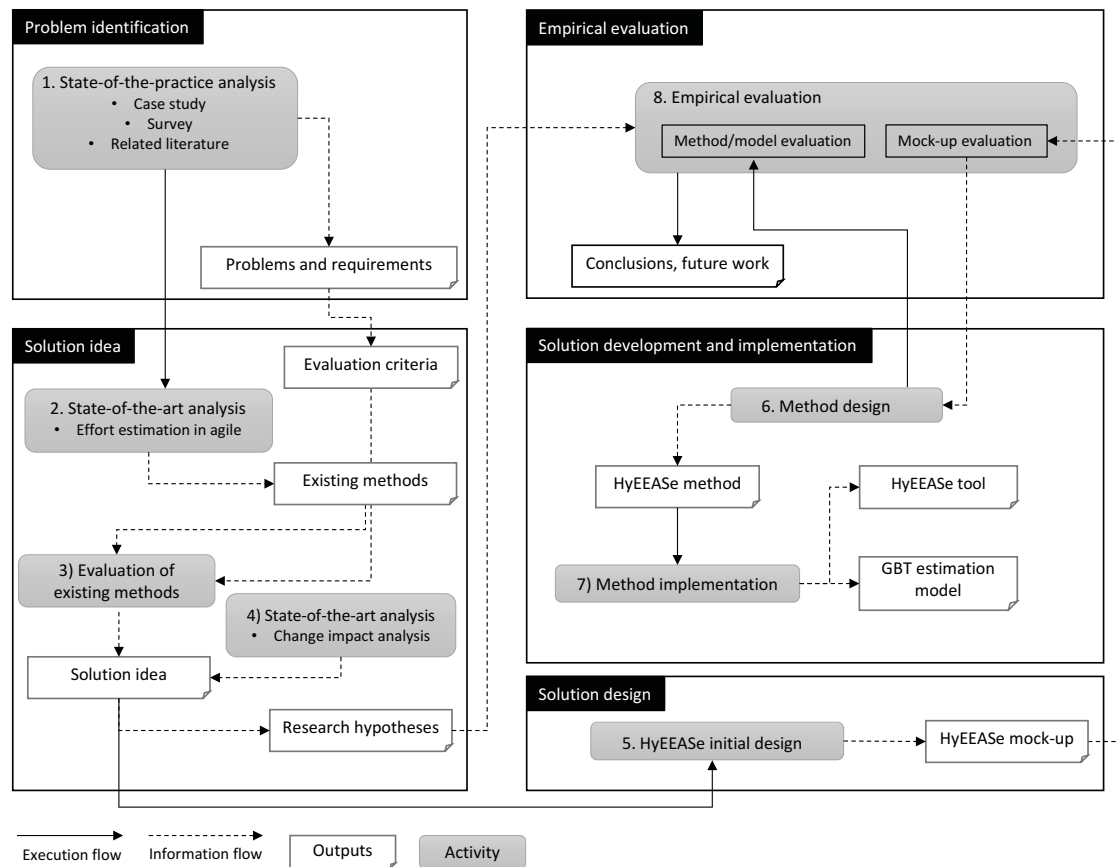


Figure 1.1: Research approach

of changes before actually implementing it [73]. The thesis aims to leverage this claimed benefit of change impact analysis for effort estimation. An additional literature review was undertaken to identify existing change impact analysis methods that could be integrated with the estimation method and would provide useful impact information supporting experts during estimation.

3. **Solution design, development, and implementation:**
To address the identified gaps, a tool based method called HyEEASe (hybrid effort estimation of change in agile software development) in close collaboration with industry was designed and developed incrementally. The method combines expert judgment with impact information and thus actually demonstrates the previously stated benefit of change impact analysis in an agile development context. Additionally, a Gradient Boosted Trees (GBT) based estimation model was also developed.
4. **Empirical evaluation:**
HyEEASe was evaluated against the research hypotheses in two industrial case studies and through a controlled experiment. The GBT based estimation model was evaluated through comparison against Agile COCOMO II.

1.4 Research problem

Effort estimation has traditionally supported the software industry for management activities like planning projects, resources and budget with estimation accuracy being the fundamental criteria for its acceptance and application.

Today, practitioners require estimation methods to be comprehensive and generate more information and feedback that helps them learn and reflect on the estimation. With this insight, they can better negotiate project budget, schedule as well as functionality. They also require estimation methods to be less complex, promote communication and knowledge sharing among teams, with minimum overhead and being accurate at the same time.

The analysis of state-of-the-art of effort estimation in agile software development reveals that the research community fails to meet the above-mentioned expectations from industry. The review further shows that the most common method in practice in this context relies heavily on human judgment [87]. Expert-based estimation methods have several limitations as they are based on limited information (subjective opinion), hampered by wishful thinking as they are prone to human judgment bias (individual and group effects) [63]. A secondary study [88] reviewed the literature on estimation in agile, iterative, and incremental projects. It concluded that there is a lack of research on finding the impact of properties of historical/current project data on estimation results. Furthermore, it has established that more empirical validation of estimation models is required. Another secondary study [113] reviewed the literature on estimating the effort in an agile context. The study concludes that the use of expert-based assessments is dominant and that there is a lack of evidence on measuring the prediction accuracy of proposed estimation techniques. It recommends considering factors other than size, for estimating effort.

With the research done in the context of this thesis, it is further recognized that existing expert-based estimation methods do not objectively consider the potential impact of a change on existing software that contributes to the effort overhead. This is because firstly, this impact information is not explicitly provided to the experts. Secondly, it is neither well integrated with expert-based estimation methods, nor is there any guidance to integrate impact analysis while doing the estimation. As a result, experts remain unaware of the impact factors affecting the software and influencing the estimates. For example, impact factors that influence estimation like code churn (no. of affected classes/methods, size lines of code), code complexity and coupling are not visible during estimation and thus inhibit experts from learning about the impact on estimation. This indicates that subjective estimation does not support systematic learning. It also makes the acquired estimates less reliable [108].

The analysis of the state-of-the-practice also revealed a complete reliance on the subjective opinion of practitioners for effort estimation. Estimates produced by these methods are often inaccurate as they are made with limited implicit knowledge leading to underestimation. In addition to information like the developer's knowledge and experience, experts recognize the need and importance of more objective information [62] as well as a means to visualize this information [106]. This includes the likely impact of a change on the existing software or historical data from previous iterations of the product [106].

This leads to the following main problem:

Comprehensive, reliable and systematic support is missing in expert-based estimation methods. The scientific and practical sub-problems related to this main problem are as follows:

1. P1: Low reliability of estimation outputs.
2. P2: Limited informative power of the estimation method.
3. P3: Subjective estimation does not support systematic learning.

In conclusion, first, there is a need for systematically storing historical data, including estimates together with the implemented change and contextual information, in an experience base for learning and reuse purposes. Second, there is a need for systematic tool support for analyzing the potential impact and required effort of changes. Thus, considerable improvement potential exists concerning systematic effort estimation in this environment and marks the main contribution of this research to the body of knowledge.

1.5 Goals and hypotheses

This section describes the goals and corresponding hypotheses.

1.5.1 Goals

The main goal was to propose and evaluate a systematic hybrid effort estimation method that increases estimation reliability, enables learnability and increases the informative power of expert-based estimation by utilizing and integrating existing change impact analysis techniques, i.e.:

1. G1: To increase the estimation reliability, i.e., increasing the accuracy of produced estimates and reducing human bias.
2. G2: To increase the informative power of the estimation method, i.e., by providing explicitly impact related useful information that will support experts in doing effort estimation. Useful here refers to the extent to which the information provided by the estimation

method is perceived as understandable and complete by the experts when performing effort estimations.

3. G3: To improve learnability, i.e., by enabling learning (awareness) of the impact factors. Learnability here refers to the extent to which the experts performing effort estimations are aware of the factors influencing estimations. Also, to enable feedback across multiple iterations to learn about discrepancies between estimated and actual effort.

1.5.2 Hypotheses

The corresponding hypotheses of the proposed hybrid estimation method (HyEEASe) were as follows:

1. H1: HyEEASe increases the estimation accuracy. HyEEASe is expected to produce estimates that are more accurate compared to pure expert-based estimation methods.
2. H2: HyEEASe reduces estimation bias. HyEEASe is expected to reduced estimation bias compared to pure expert-based estimation methods.
3. H3: HyEEASe provides useful (understandable, complete) impact information of the potentially impacted code. HyEEASe is expected to provide understandable and complete impact related information to the experts during estimation as compared to pure expert-based estimation methods.
4. H4: HyEEASe enables learning (awareness) of the impact factors. HyEEASe is expected to increase the awareness of the experts about the code-based impact factors (like impact size and complexity that would affect the existing system) while estimating effort for a change request as compared to pure expert-based estimation methods.
5. H5: HyEEASe enables learning of associated effort (from sprint to sprint). HyEEASe enables feedback across multiple iterations to learn about discrepancies between estimated and actual effort.

Figure 1.2 shows a mapping of the research problems to goals and hypotheses.

1.6 Evaluation strategy

Different hypotheses of the research have been evaluated using different strategies. Figure 1.3 shows a mapping of research goals, hypotheses, and corresponding evaluation strategies:

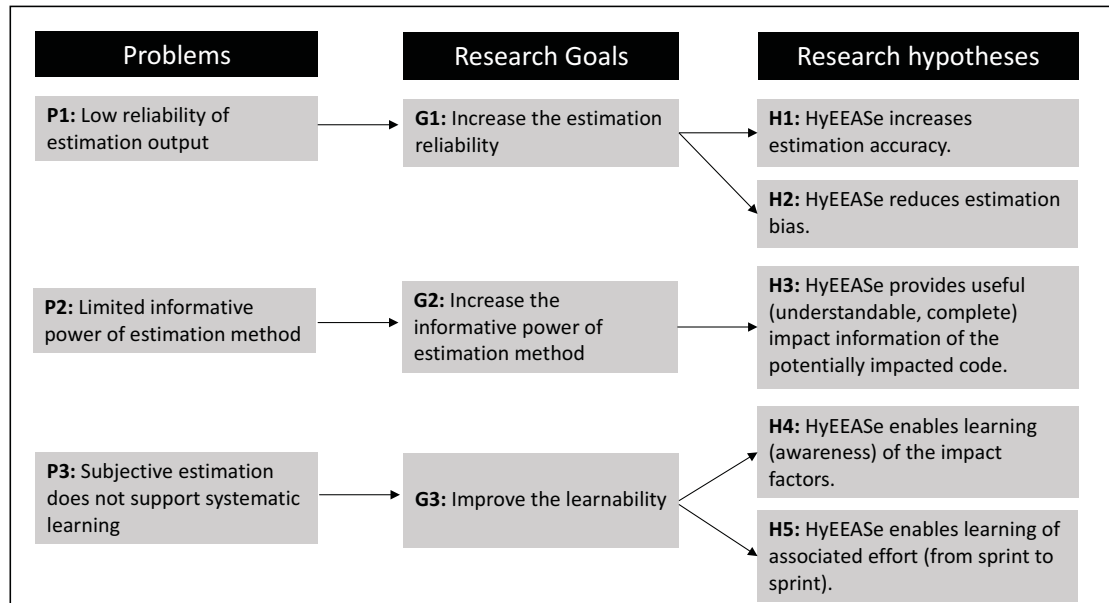


Figure 1.2: Mapping of problems to goals and hypotheses

1. Case study at SAP SE: The objective of this case study was to evaluate the concept of utilizing change impact analysis for effort estimation. Furthermore, through mock-ups, the perceived usefulness (H3) of HyEEASe was evaluated.
2. Survey at Insiders Technologies: The objective was to evaluate the effectiveness (accuracy (H1), bias (H2)) as well as the perceived usefulness (H3) and learnability (H4) of the refined HyEEASe with tool-support.
3. Controlled experiment with students at TUKL: The experiment was conducted to find the usefulness (H3) and learnability (H4) of the hybrid method in comparison to the pure expert-based estimation method.
4. Data based evaluation at Insiders Technologies GmbH: The objective was to evaluate the estimation effectiveness (accuracy (H1), bias (H2)) of the GBT estimation model by comparing it with Agile COCOMO II estimation model [33].

1.7 Background

To utilize change impact analysis for supporting effort estimation in an agile development context, the related research areas, i.e., effort estimation methods and change impact analysis techniques were explored. The following subsection gives a summary of background information in these fields.

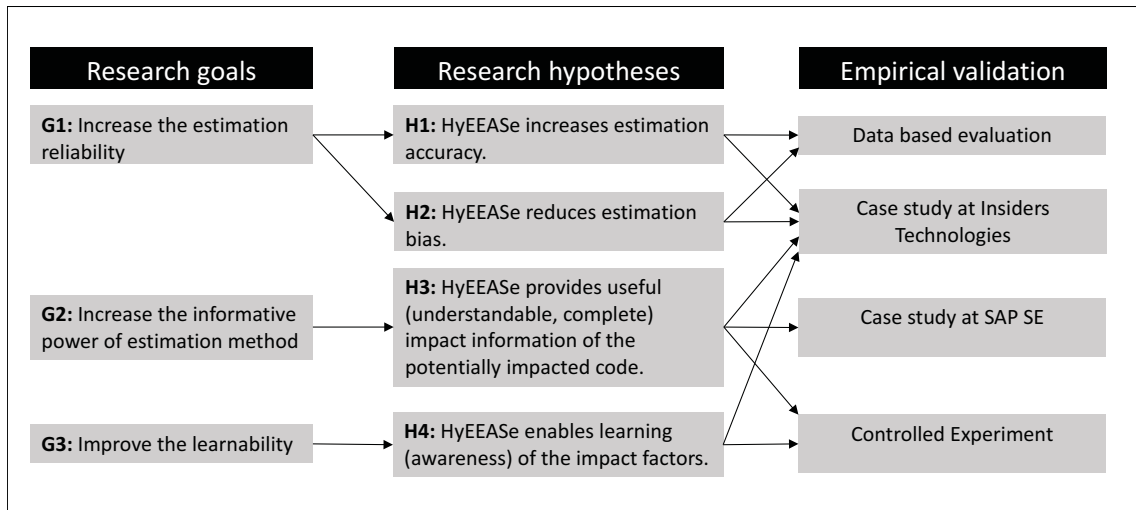


Figure 1.3: Empirical validation strategies

1.7.1 Effort estimation methods

In traditional software development, many effort estimation methods and models have been proposed in research. These may be classified as [112]:

1. Data-driven (model-based, memory-based and composite) methods, like COCOMO I.
2. Expert-based methods like Wideband Delphi, Planning Game, Analytic Hierarchy Process.
3. Hybrid methods like Bayesian Belief Nets (BBN) and CoBRA¹[38].

Data-driven estimation methods are often models that are fixed (e.g., COCOMO family) which can only perform better on the type of data they have been developed on. They are incapable of adapting themselves to the agile context, cannot handle uncertainty and therefore cannot improve the effectiveness of estimation especially when applied to a new data set without prior calibration[84, 112]. Hybrid estimation methods, are either inherently too complex (e.g., BBN) or are too effort-intensive (e.g., CoBRA) to be deployed and used in an industrial setting especially in agile development context [112].

Each of these methods claims to have addressed a problem in effort estimation. However very few of them have demonstrated the claims in industrial settings. Also, very few individual studies are found that address the effort estimation specifically in the agile context. Expert-based methods are found to be the most used estimation method in agile context [113], but their estimation accuracy is hampered by inconsistencies

¹CoBRA is a registered trademark of the Fraunhofer-Gesellschaft and stands for Cost Estimation, Benchmarking and Risk Assessment.

and wishful thinking [63]. However, due to the lack of evidence that model-based methods like COCOMO produce more accurate estimates than expert judgment, the use of the latter approach is widespread [62].

In the context of this thesis, a literature review was made where effort estimation techniques used in an agile context were further analyzed for estimation accuracy, usefulness, and learnability. It was found that none of the existing estimation methods (whether in traditional development in general or in agile development in particular) so far have considered the quantification of the impact that a change has on existing software artifacts. Further, explicit consideration of the most relevant effort drivers is also not addressed in expert-based methods in particular. Furthermore to collect and analyze these effort factors a huge amount of data and cost are required in data-driven or model-based methods like COCOMO [112]. None of the existing estimation methods so far have directly integrated results from an impact analysis.

1.7.2 Change impact analysis

Change impact analysis is a technique that identifies the effects of a change or estimates the tasks required to implement a change [73]. It can be used for understanding a program, predicting the impact of a change and estimating the costs of change before implementing it. However, none of the studies especially in the agile context have used this concept to provide support to experts during the estimation process. A secondary study on change impact analysis [73] has identified 23 unique techniques that are broadly based on dependency analysis.

In the context of this thesis, another literature review was made where these and other techniques were further analyzed for their support in estimating the effort of a change by identifying its potential impact on the existing system. It was found that two main types of analysis exist: dynamic analysis and static analysis each with subtypes. Subtypes of dynamic IA include offline and online analysis. Subtypes of static analysis are:

1. Structural static analysis: It focuses on static analysis of the structural dependence of the program and construction of the dependency graph. Knowing the structural dependence allows predicting which elements are impacted based on this dependence.
2. Textual analysis: It extracts conceptual dependence (coupling) based on the analysis of the comments and/or identifiers in the source code.
3. Historical analysis: It is performed by mining information from multiple evolutionary versions of the software in software repositories.

Software repositories can also provide information on the co-change coupling to predict future changes. In the context of this thesis, the

structural static analysis and historical analysis were combined for supporting expert-judgment in agile development.

1.8 Research contribution

The contributions of this thesis are as follows:

1. The development of a hybrid estimation method in the context of agile software development by combining change impact analysis and expert judgment.
2. In pursuit of developing the method, several other scientific and empirical contributions were made i.e.
 - Development of key elements of the method: 1) Analysis of affected software concerning the impact of the change. 1.1) Providing historical estimates and impact information for similar impacted parts. 2) Measurement of size and complexity of change. 3) Visualizing the impact information for the experts to interact with. 4) Providing an experience base to store impact and estimation data for organizational learning and future reuse. 5) A prototype tool support.
 - Literature reviews were conducted for establishing the state-of-the-art for change impact analysis and effort estimation in agile development.
 - Industrial case study and a survey using interviews and questionnaires were conducted for establishing the state-of-the-practice of estimation methods at the case companies, i.e., SAP SE and Insiders Technologies GmbH respectively.
 - The development of a prototype tool to support the method as well as the GBT estimation model are among other scientific contributions.
3. The evaluation of HyEEASe also involved several empirical contributions including:
 - Two case studies at two companies were conducted. The first case study was performed at SAP SE, a German multinational software corporation, to evaluate the perceived usefulness and effectiveness of the concept through mock-ups. The second case study was done at Insiders Technologies GmbH, where the effectiveness, the perceived usefulness, and the learnability of the refined hybrid method (HyEEASe) through the use of a prototype tool was evaluated.
 - A controlled experiment was conducted with the BS and MS students taking Software Process and Project Management

(SPPM) course at the Department of Computer Science, Technische Universität Kaiserslautern (TUKL) to find the usefulness and learnability of the proposed hybrid method (HyEEASe).

1.9 Thesis structure

The thesis is structured as follows:

- Chapter 2: State-of-the-practice: This chapter gives an overview of the current estimation methods practiced in an agile development context. It further identifies the problems in prevailing estimation methods as well as elicit the requirements for an improved estimation method. The overview is based on a review of the related literature and results of an industrial case study and a survey.
- Chapter 3: State-of-the-art: This chapter describes the literature reviews of a) existing effort estimation methods in agile development, b) existing change impact analysis techniques proposed in the published literature. Based on an analysis of the impact analysis techniques, the techniques that could be integrated into expert-based estimation and would provide necessary impact information to support experts during estimation were identified.
- Chapter 4: HyEEASe - Method for effort estimation in agile development: Based on the results of Chapter 3, a systematic hybrid effort estimation method is developed that integrates impact analysis in expert judgment. Additionally, a GBT based estimation model is also developed. The detailed solution idea, design and implementation are described in this chapter.
- Chapter 5: Empirical evaluation: Different research hypotheses have been evaluated using different strategies, i.e., the reliability, perceived usefulness and learnability of the developed hybrid method are evaluated with the case companies. A controlled experiment was conducted to evaluate the usefulness and learnability of the hybrid method. The GBT model is evaluated through comparison with a research alternative, i.e., Agile COCOMO II. This chapter describes the design and results of these empirical evaluations.
- Chapter 6: Conclusions: This chapter summarizes the results of the thesis work and draws directions for future research.

2 State-of-the-practice

2.1 Introduction

This chapter describes the state-of-the-practice of estimation methods. It characterizes the methods based on their accuracy and the information required by the experts to increase their usefulness. The industrial perspective was captured by reviewing published literature reporting state-of-the-practice and through a case study and a survey conducted as part of this thesis.

Performing a case study and survey with case companies, it was investigated why and how estimation is done and what factors are considered relevant for making estimates and for improving the reliability of estimation output. Furthermore, the issues in prevailing estimation methods were identified as well as the requirements for an improved estimation method were elicited. Two studies are presented in this chapter in Section 2.3 and Section 2.4 respectively. Both studies shared the same goals. The results of the studies include issues related to the estimation methods. These issues and the issues identified through the related literature were then compared. The findings were used to formulate the requirements of estimation methods.

The research done for this chapter was conducted as part of research projects, Abakus, grant number 01IS15050G, and HyEEASe, grant number 01IS12053 funded by the German Ministry of Education and Research (BMBF). The results were also published, details can be found at [107].

This chapter is structured as follows: Section 2.2 highlights the related literature concerning issues related to estimation methods. Section 2.3 and Section 2.4 detail the design, execution, results of industrial case study and the survey.

Section 2.5 identifies the problems of expert-based estimation methods that will be addressed by this thesis. Section 2.6 reports a set of requirements that need to be fulfilled to address the problems of the estimation methods. Section 2.7 summarizes the chapter.

2.2 Related literature

A few secondary studies exist in the published literature that establish the state-of-the-art regarding different aspects of effort estimation in the context of agile development.

One secondary study [88] reviewed the literature on estimation in agile, iterative, and incremental projects. It concluded that there is a lack of research on finding the impact of properties of historical/current project data on estimation results. Furthermore, it has established that more empirical validation of estimation models is required.

Another secondary study [113] reviewed the literature on estimating the effort in an agile context. The study concludes that the use of expert-based assessments is dominant and that there is a lack of evidence on measuring the prediction accuracy of proposed estimation techniques. It recommends considering factors other than size, for estimating effort.

A few surveys are also found in the published literature that investigated the state-of-the-practice regarding different perspectives of effort estimation in this context. A survey [82] reported the results of interviews conducted with managers of 18 different companies and from 42 different projects in the Norwegian industry. The results showed that projects that adopted a flexible development process model experienced fewer effort overruns than the projects that adopted sequential development models. Using a dataset from [82], one study [83] established that expert estimation is the dominant estimation method, that the use of formal models does not impact estimation accuracy and that managers are convinced of their high estimation accuracy, which in reality is not that high. The latter study, however, did not distinguish between projects using traditional or agile development models.

Another survey [114] conducted with 60 agile practitioners from 16 different countries concluded that the use of expert-based assessments is dominant, with the effort being underestimated. It further reported that using a combination of estimation techniques may affect estimation accuracy positively, that estimation inaccuracy is mainly due to requirements and management related issues, and that team-related cost drivers are frequently used in agile development.

These studies have tried to establish the state-of-the-practice of effort estimation from different perspectives, but none of them has explored the estimation of the impact of change requests concerning their magnitude as well as the factors constituting effort overhead and affecting estimation accuracy.

2.2.1 Estimation methods issues

Some issues of expert-based estimation methods mentioned in the published literature are as follows [62], [63], [113], [88]:

1. Expert estimation is inaccurate and biased, where underestimation is a trend.
2. Expert estimation is based on limited information.

3. Lack of research on finding the impact of properties of historical/current project data on estimation results.
4. Lack of empirical evidence on the effectiveness of proposed estimation techniques.

It is obvious from problems 2 and 3 that expert-based estimation needs to consider more information to make informed decisions while doing the estimation.

From the review of published literature, it is recognized that existing expert-based estimation methods do not objectively consider the information, like the potential impact of a change on existing software, that contributes to the effort overhead. This is because firstly, this impact information is not explicitly provided to the experts. Secondly, it is neither well integrated with expert-based estimation methods, nor is there any guidance to integrate impact analysis while doing the estimation. As a result, experts remain unaware of the impact factors affecting the software and influencing the estimates. This leads to the following issue:

5. Subjective estimation does not support systematic learning.

Experts performing effort estimations remain unaware of the complexity and volume of the factors influencing the overall system as well as the estimations. For example, impact factors that influence estimation like code churn (no. of affected classes/methods, size lines of code), code complexity and coupling are not visible during estimation and thus inhibit experts from learning about their impact on estimation. This indicates that subjective estimation does not support systematic learning. It also makes the acquired estimates less reliable.

There is a need to provide this information during estimation that would support experts in making informed decisions and thus improve estimation accuracy and reduce bias.

2.3 Case study

This section describes the goal, sample, design, execution, results, and analysis of the case study.

2.3.1 Goal

The goal of the case study was to explore and analyze the prevailing estimation process concerning its operationalization and effectiveness in the context of agile software development from the perspective of agile development teams.

We, therefore, wanted to explore the prevailing estimation methods (purpose, process, tools used, etc.) and the issues associated with them.

2.3.2 Sample and population

The case study was performed with a German multinational software corporation, SAP SE (Systems, Applications & Products in Data Processing). SAP develops enterprise applications in terms of software and software-related services to manage business operations and customer relations [7]. To keep pace with the changing market trends and customer needs, the company has also integrated agile practices such as XP and Scrum. Since the aim was to explore the estimation process in an agile context; the focus was on the ongoing projects practicing agile principles.

The context was narrowed down to the Scrum and XP methodology and defined the case to be studied as Scrum and XP teams performing estimation while planning for a sprint. The unit of analysis was intentionally selected to allow the comparison of deviant cases such as estimation method (individual vs. group estimation) and team size (small vs. large). Accordingly, a sample of three units of analysis corresponding to three different software development teams was drawn. Table 2.1 shows the units of analysis. The experience of the teams is aggregated and is shown as the median value.

Team A comprised 10 members with an average experience of four years of agile development and 14 years of software development.

Team B comprised seven members with an average experience of four years of agile development and 18 years of software development.

Team C comprised 20 members with an average experience of five years of agile development 10 years of software development.

Scrum, test-driven development (TDD) and extreme programming (XP) are being practiced by all three teams. The programming languages used are ABAP¹ and Javascript. All teams are collocated, delivering 12 sprints per year, with a sprint lasting for four weeks.

2.3.3 Design

Using case study research, two observations and eleven interviews were conducted with three agile development teams at the case company. This section is based on the published work [106] of the author of the thesis.

¹ABAP (Advanced Business Application Programming) is a high-level programming language developed by SAP.

A multiple-case design was chosen to get more in-depth insights on how estimation has been performed in agile software development. A multiple-case study is a variant that investigates at least two cases of the same phenomenon. Thus, it is especially useful for examining a phenomenon across different settings and for exploring how the results vary among them [99]. This case study was performed and documented according to the guidelines of Runeson and Höst [96].

The case study protocol used in this research includes (1) the procedures for selecting the units of analysis and making arrangements for the field-work, (2) the informed consent for protecting the participants and the organization, (3) the data collection procedures, forms, and plan, and (4) the data analysis procedures. The materials can be found in Appendix A.

Table 2.1: Sample

Characteristics	Team A	Team B	Team C
Team size	10	7	20
Team experience with software development (in years)	14	18	10
Team experience with agile development (in years)	4	4	5
Agile methods being used	Scrum, TDD, XP	Scrum, TDD	Scrum, TDD, XP
Programming language	ABAP	Javascript	Javascript

2.3.4 Data collection

The case study was performed during the second half of 2014. To reduce bias and increase the credibility of the empirical data and findings; multiple data sources were used following a triangulation approach.

The first source driving the qualitative analysis was observation sessions of sprint planning meetings to watch the effort estimation process.

The second source was a set of interviews. The materials can be found in Appendix A. Data collection was performed as follows:

1. **Introduction meeting:** A meeting was held to introduce the purpose, process, and outcomes of the case study to the Scrum masters (SM) and product owners (PO) of the three selected teams. All questions and concerns were resolved. Informed consent was collected, observations and interviews were scheduled.
2. **Observations:** Two researchers observed the sprint planning meetings of the teams. They independently recorded data regarding the participants (roles and responsibilities), their interactions, the estimation process (inputs, steps, rationale, and outputs), and the

tool used. After each observation, they compared their notes and generated a consensual observation protocol.

3. Interviews: Interviewees were selected so that the main roles in the team were covered, i.e., PO, SM, and at least one developer be interviewed. However, the number of interviewees varied according to the team size. In Team A and B, there were three each, and in Team C, there were five interviewees.

The design of the interview consisted of three main parts, and the duration of the interviews was 45 minutes to one hour. In the first part of the interview, an introduction of the study goal was provided, data confidentiality was ensured, and demographic questions were asked. In the second part, the interviewees were asked about their experience in general and with the agile development approaches in particular. They were also asked about their role and responsibilities at SAP and their working domain, platform, language and effort estimation concepts in general. In the third part, questions related to effort estimation, in particular, were asked in a semi-structured interview.

Altogether, 11 interviews were performed either face-to-face or via video conferencing. Two researchers were involved in conducting the interviews. One performed the interview, and the other took notes. All interviews were recorded and transcribed. After each interview, they compared their notes with the electronic record and generated a consensual interview protocol. The interview questionnaire was derived by operationalizing the research questions using the GQM approach [115].

2.3.5 Data analysis

Observations and interviews: Qualitative data analysis was performed by using coding techniques and making use of the tool, MAXQDA [6]. Quantitative data analysis was performed using IBM SPSS Statistics [10]. Descriptive statistics including the sample median (Mdn), minimum (Min), maximum (Max), frequencies, and valid percentages are reported. For five-point rating scale data, one-sample Wilcoxon signed-rank test was used to test for significant differences from the midpoint ($H_0: \text{Mdn}(x) = 3$) [119]. In these cases, the observed value (Z) and the significance level (p) has been reported.

Feedback: The results were aggregated, presented, and discussed with the corresponding teams. This approach was used to increase the trustworthiness of the results and ensure that they reflect the participants' insight. The participants reviewed the results and agreed with the findings.

2.3.6 Threats to validity

In the following, we will discuss some threats and the ways we tried to mitigate them.

- **Construct validity:** To strengthen the empirical evidence, multiple data sources (e.g. interviews and observations) were used following a triangulation approach. Additionally, another experienced researcher reviewed the design protocol and provided his expert opinion on how to conduct the studies. This feedback also enhanced the quality of the study design.
- **Internal validity:** To avoid bias in study results, only those teams were approached that had sufficient experience in agile methodologies. During the interviews, the practitioners were asked about their experience specifically in agile development in addition to their experience in software development in general. To avoid evaluation apprehension, the practitioners were assured of the anonymity of their personal and organizational data. Furthermore, we faced slight deviations from the data collection protocols due to time constraints and low staff turnover. We mitigated this threat by conducting the feedback session where the teams reviewed the study results and agreed with the findings.
- **External validity:** As our case study used a convenience sample, our results might not be generalizable. However, they may be repeatable in a similar context and with similar characteristics concerning agile development.
- **Conclusion validity:** We aggregated the results of the teams and only used median values, and the number of respondents to identify common practices/views and to point out potential future research areas.
- **Reliability:** Through involving more than one researcher in each step of the studies' design, review, and execution and using the defined protocol for reducing bias and increasing the credibility of the results, we mitigated this threat.

2.3.7 Results - estimation method

This section describes the results of the study in terms of detailing the estimation method.

- **Estimation purpose:** The teams do estimation for scheduling user stories but not for risk assessment [56], project bidding, or budgeting [38] for example.

Other than that, the teams perform estimation to develop a common understanding among them regarding is what needs to be done in a sprint.

- Estimation method: Teams A and B do individual estimations, whereas, Team C plays the estimation game [9] where estimates are provided through consensus.
- Estimation unit: Teams A and B estimate effort in Person hours, whereas, Team C estimates complexity of BLI in Story points.
- Tools used: The teams use MS Excel and the issue tracking and project management tool, JIRA [11].
- Estimation process: It was observed that the teams have a properly laid out process for organizing estimation sessions where the whole team participates. All the teams do estimation in Scrum sessions at the sprint planning level. The estimates are made for requirements at the granularity level of backlog items (BLIs) which is a fine-grained level of a User Story (US).

2.3.8 Results - estimation method issues

From observations and interviews, the identified issues are as follows:

1. Effort is underestimated. A difference of opinion was observed among the teams. The participants of all three teams at management level were convinced that underestimation is a trend whereas the team members of teams A and B were unsure about the accuracy. Team C perceived their estimates are reasonably accurate. It is because the team members think that the discussion they make during the planning helps them develop a common understanding of BLIs to be done in a sprint. However, on the contrary, it was observed that no discussion was made during BLI estimation. Some participants remained unaware of why a certain BLI is given a certain estimate. That is, they cannot guarantee whether they have gotten a common understanding of BLIs and therefore, this leaves room for inaccuracy.
2. Lack of measurement data (i.e., data on estimates and actuals is not stored). Concerning the previous problem, it was also observed that the estimates made and the actual effort spent is only available till the end of a sprint. They are, however, neither stored nor used to measure effort estimation accuracy. There is a lack of basis for assessing and improving estimation performance (accuracy).

Though the teams are convinced about their high estimation performance although no objective evidence is available. Since the impact of estimation performance on project performance (e.g., schedule slippage) is unknown, this prevents the PO from making

commitments to external stakeholders regarding BLI delivery in a planned sprint.

3. Estimation is based on partial information for BLI estimation. It was observed that factors with significant impact on effort estimation are either considered to a limited extent or not considered at all. It is because the current tools do not provide this information and also are incapable of generating it. It is explained below:

To investigate what factors are considered during estimation, we captured the interviewees' perspectives in different ways. First, the interviewees were asked in an open-ended question to provide factors they consider (implicitly in their minds) while making estimates. Figure 2.3 shows the factors with the number of practitioners providing them (first column). "Developer implementation experience" was considered most relevant (provided by six out of 11 practitioners) followed by Developer knowledge" and "Capacity". According to the teams, "Capacity" means the amount of workload a certain developer has. However, the factor "Estimates made for a similar BLI in previously completed sprints" does not seem relevant for making estimates as no one mentioned it.

Second, the interviewees were presented with seven factors (see Table 2.2) to rate the extent to which each factor is relevant for the estimation process (scale: five-point ratio scale, with 5 being very relevant and 1 not being relevant at all). There was also an opportunity to provide additional factors they thought were relevant and rate them accordingly. However, no one provided any additional factor. Table 2.3 in the second column, shows the aggregated results of the teams. Almost all factors are rated as relevant to be considered for the estimation process. However, the factor "Estimates made for a similar BLI in previously completed sprints" does not seem relevant for making estimates (median= 2).

Finally, the interviewees were presented with the same seven factors (see Table 2.2) to rate the extent to which each factor has the potential to improve estimation accuracy (scale: five-point ratio scale, with 5 being very relevant and 1 not being relevant at all). There was also an opportunity to provide additional factors they thought could improve accuracy and rate them accordingly. However, no one provided any additional factor. Table 2.3 in the third column, shows the aggregated results of the teams. Here again, almost all factors were rated as having a high potential to improve estimation accuracy. However, the teams were unsure whether having estimated similar BLI in the past (i.e., the factor "Estimates made for a similar BLI in previously completed sprints" has any effect on improving estimation accuracy.

During the analysis, it was found that factors that are rated as considered highly relevant for making estimates are either considered

to a limited extent or are not considered at all during the estimation process. For example, in Table 2.3, the factor “BLI impact on parts of the developed system” is rated as relevant whereas only four out of 11 practitioners are implicitly considering it. Similarly, the factors “Developer’s experience with making estimates” and “BLI complexity” are also rated high for improving estimation accuracy but are being implicitly considered to a limited extent.

On the other hand, the factor “Dependencies among BLI” is rated as relevant, whereas no one is implicitly considering it. It can also be noted that it is also rated as high for improving estimation accuracy.

This shows that when provided with a factor explicitly, the practitioners realized this factor’s relevance and impact on accuracy. Such explicit consideration of factors is missing in the currently used estimation methods.

However, the factor “Estimates made for a similar BLI in previously completed sprints” is neither rated as relevant (median = 2) nor any participant implicitly considers it as well as the practitioners were unsure (median value = 3) of the impact of this factor on effort estimation accuracy as seen in Table 2.3. One possible reason for this could be the practice of not storing estimates. Therefore, no one realizes the impact it can have on estimation accuracy.

2.4 Survey

This section describes the survey goal, design, execution, results and discusses them.

2.4.1 Goal

This study shares the same goal as presented in Section 2.3.

2.4.2 Sample and population

The survey was performed with a German software company, Insiders Technologies GmbH [3]. The domain of Insiders Technologies is mainly information systems. It is a product development company that develops document management solutions for the public, insurance, commercial, and finance sector. It focuses on the development of software for processing, extracting, and classifying information from any kind of business correspondence for the insurance domain. The used programming languages are C++, Javascript, and HTML. The releases have been developed using Scrum since 2009. The company follows agile software development processes including Scrum and XP. Altogether three interviews were conducted with one development

Table 2.2: Factor definition

Factor name	Factor definition	Factor category
Developer's experience with making estimates.	Refers to the experience of a developer with estimating development activities.	People-based impact factor.
Developer's knowledge of BLI.	Refers to the knowledge of a developer regarding the system/component he/ she is working on as well as a certain BLI to be implemented in that system/component.	People-based impact factor.
Dependencies among BLI.	Refers to existing dependencies (coupling with other BLIs) among the BLIs to be implemented in a sprint.	Code-based impact factor.
BLI complexity.	Refers to the complexity of a new BLI in terms of its type, i.e., functional/non-functional, number of required inputs/outputs, internal/external interfaces, design constraints, user/location, and required feature [103].	Code-based impact factor.
BLI impact on parts of the developed system.	Refers to the impact of implementing a new BLI on the existing system, i.e., identification of system classes/ components affected by the change or the number of test cases required, for retesting the system after a change, etc.	Code-based impact factor.
Estimates made for a similar BLI in previously completed sprints.	It means a similar (in terms of actual and estimated effort) BLI was already estimated and implemented in the past, so both it's 1) actual effort and 2) estimation error should be known.	Data-based impact factor

team comprising eight members. The roles of participants were development manager, product owner (PO) and a developer. The average experience of the participants was more than seven years in software development and more than five years in agile development.

2.4.3 Design

Through three interviews along with questionnaires were conducted with one agile development team at the case company. The study was designed using pre-defined protocols for questionnaires and interviews. The materials can be found in Appendix B. The protocol comprised of sending a formal invitation to the development team, introducing the questionnaire/ interviews describing the goals and procedures and guidelines, e.g., how to provide answers to the questions and the time required to conduct interviews.

The questionnaire comprised of two sections. The first section contained questions related to estimation scope and context information and demographic information. The second section comprised of few closed-ended questions related to estimation practices. The interview design was an elaborated version of the initial questionnaire and con-

Table 2.3: Factors comparison

Factor name	Implicit consideration of factors (No. of participants)	Explicit consideration of factors (factor rating - median)	Factor importance for improving estimation accuracy (factor rating - median)
Developer's experience with making estimates	4	4	4
Developer's knowledge of BLI	5	5	5
Developer's experience of implementation	6	5	4
Dependencies among BLI	0	4	4
BLI complexity	4	5	4
BLI impact on parts of the developed system	4	4	4
Estimates made for a similar BLI in previously completed sprints	0	2	3
Capacity	6	0	0

centrated mainly on participant's experience of the estimation method, the strength, and weaknesses of the method in practice, as well as the identification of requirements regarding effort estimation.

2.4.4 Data collection

Multiple data sources were used to increase the credibility of the data and findings. The data were collected using a questionnaire, the first source, followed by interviews, the second source, as data collection instruments. More than one researcher was involved in designing, conducting as well as analyzing the collected data. The materials can be found in Appendix B. Data collection was performed as follows:

1. **Off-line Survey:** Invitation to the survey was sent initially to the company. The invitation stated the goals of the survey and contained few closed-ended questions regarding the demographic information, organization context, product characteristics, estimation scope and context and few open-ended questions regarding characteristics of the current estimation practices.
2. **Face-to-face interviews:** After getting the response to the off-line survey, the invitation to the interviews was sent to the development team. The invitation stated the goals of the interview and contained open-ended questions regarding the characteristics of the current estimation practices.

Altogether three interviews were conducted at the case company. Each interview lasted about 45-60 mins. The interviews were con-

ducted face to face by two researchers in the setting of a focus group. One researcher interviewed whereas the second took notes. The participants were ensured that the information they provide would be handled confidentially. Interviews were recorded. After each interview, both researchers compared notes with the electronic data to reach a consensus. Finally, the interview transcripts were revised by the interviewees.

2.4.5 Data analysis

Interviews and questionnaire: The same tools and techniques were used as mentioned in the previous case study in Section 2.3.5. The qualitative data analysis was performed using coding techniques. The analysis was performed by more than one researcher to increase the credibility of the results. Only the aggregated results are reported.

Feedback: The aggregated results were presented and discussed with the case company. The participants reviewed and agreed with the findings. This feedback increased the trustworthiness of the results and reflected the company's insights.

2.4.6 Threats to validity

This study shares the same threats as mentioned in Section 2.3.6.

2.4.7 Results - estimation method

This section describes the results of the study in terms of detailing the estimation method.

- **Estimation purpose:** The company does effort estimation mainly for project planning and for creating business value, i.e., creating new features for making new business markets or using/ refactoring the existing features for maintaining the existing customers. Their main goal does not seem to focus solely on creating attractive offers/ quotations for customers, but they need effort estimation for planning their business and project goals.
- **Estimation method:** Planning poker [5] is the prevailing estimation method. Estimation is done for the granularity level of "Stories/ User stories".
- **Estimation unit:** complexity (in Story points - SP) is estimated.
- The whole team participates in Planning poker during Scrum session, but the only developer gives her/ his estimate for her/his user story which is then discussed in the team.

- Tools used: A company-specific tool is being used for documenting and using estimates for future reference.
- Estimation process: The case company uses a multistage process to estimate a user story as follows:
 - Package grooming: In this stage the team gives an initial rough estimate for a user story so that the product owner (PO) can prioritize the user story in the backlog. After that, the user story is included in a future sprint.
 - Grooming: In this stage, the requirements for the user story are refined by at least two members of the development team together with the PO.
 - Research: In this stage the user story is researched by at least two members of the development team and is broken down into low granularity sub-stories, called “tasks”, which can be implemented independently from each other.
 - Q&A: In this stage, the developers present the user story to the rest of the team to re-estimate the whole story more precisely using planning poker. The team then divides the estimate of the story into its corresponding tasks based on the associated complexity and risk.

2.4.8 Results - estimation method issues

From interviews, the identified issues of estimation methods are as follows:

1. Estimation methods used in the company do not provide accurate estimates where underestimation being a trend. Inaccurate estimates result in project delays, time/ budget overruns, dissatisfied customers and employees alike, loss in businesses.
2. Estimation is based on individual judgment with limited information. Thus, the estimation becomes difficult when for example, the necessary experience/ required expert is not available.
3. Lack of standardization (of process and documentation). There is no systematic process to learn about the factors influencing estimates as well as the impact on the overall system.
4. No mechanism of reusing data from similar user stories implemented in the past sprints for making estimates for the new ones. Unlike the case study, this company does have some measurement data however it is not being used during estimation.

2.5 Estimation method issues - literature, case study, and survey

In this section, the issues of expert-based estimation methods identified by both the studies and the related work are compared and discussed. Table 2.4 shows the issues identified by both the academia (related literature) and industry (case study and survey).

Table 2.4: Identified issues of estimation methods

Identified issues	Source
Estimation inaccuracy	Both academia and industry
Effort underestimation	Both academia and industry
Estimation based on limited information	Both academia and industry
Lack of research on finding the impact of properties of historical/current project data on estimation results	Academia
Subjective estimation does not support systematic learning	Both academia and industry
Lack of evidence on measuring estimation accuracy	Academia
Lack of standardization (of process and documentation)	Industry
No mechanism of reusing data from similar user stories implemented in the past	Industry

Out of all the issues presented in Table 2.4, following are those common issues of expert-based estimation methods that will be addressed as “problems” by this thesis (along with their corresponding goals):

1. P1: Low reliability of estimation output. Estimates generated by the expert-based estimation methods employed in both the companies are inaccurate with underestimation being a trend. Due to this problem, project planning and controlling are gets affected. The management cannot make commitments to the external stakeholders which further affects project bidding adversely.

G1: The corresponding goal of the thesis is to increase the estimation reliability, i.e., increasing the accuracy of produced estimates and reducing human bias.
2. P2: Limited informative power of the estimation method. Expert estimation is based on partial information that may exist in the minds of experts. Such information is useful only when explicitly provided and in an understandable form to support experts while doing the estimation. For example, currently, the factors with a significant impact on effort estimation are either considered (implicitly) to a limited extent or not considered at all. Due to this problem, the estimation becomes difficult when the necessary experience/ required expert is not available.

G2: The corresponding goal of the thesis is to increase the informative power of the estimation method, i.e., by providing explicitly impact related useful information that will support experts in doing effort estimation.

3. P3: Subjective estimation does not support systematic learning. Expert-based methods lack a mechanism to systematically storing estimates together with the implemented change and contextual information in an experience base for learning and reuse purposes. Due to this, experts performing effort estimations remain unaware of the complexity and volume of the factors influencing the overall system as well as the estimations.

G3: The corresponding goal of the thesis is to improve learnability, i.e., by enabling learning (awareness) of the impact factors.

2.6 Requirements of estimation method

Table 2.5 describes a set of requirements that needs to be fulfilled to address the problems of the estimation methods. These requirements were gathered based on the results and our experience of conducting state-of-the-art and practice analysis.

Certain other expectations (wishes instead) were also observed during state-of-the-practice analysis. For example, an improved estimation method should automatically calculate velocity (speed) and story points per defect based on historical data. It should be able to reuse data from similar user stories implemented in the past and derive standard estimates for certain user stories.

2.6.1 Mapping of requirements to problems and goals

In this section, a mapping of the requirements described in Table 2.5 to the problems and goals of the thesis is provided (see in Table 2.6). It is observed that to address a problem of estimation methods, a subset of requirement needs to be fulfilled, as explained below:

1. P1: Low reliability of estimation output.

Requirements: To overcome this problem, an improved estimation method is required that increases accuracy and reduces bias. The method should generate reliable yet easy to analyze and easy to maintain estimates. Such a method should provide all necessary and useful information required by the expert to make less biased estimates.

2. P2: Limited informative power of the estimation method.

Table 2.5: Requirements regarding estimation method

ID	Requirements	Definition
R1	Reliability	The estimation method should produce reliable (accurate and precise) estimation outputs.
R2	Complexity	The estimation method should be easy to understand and apply. It should provide outputs that too are easy to analyze, understand and maintain.
R3	Informative power	The estimation method should have all necessary useful (understandable and complete) information. This information will help experts in making accurate estimates. This information includes the explicit provision of code-based impact related factors that influence estimates, i.e., factors like the complexity of the story, the impact of the user story on the overall system.
R4	Systematic learning support	The estimation method should systematically make experts aware of the impacts of implementing a user story. With this awareness, the experts would learn about the effects of impact factors and thus would make informed decisions regarding estimates.
R5	Tool support	The estimation method should be tool supported. The tool should be interactive, easy to use, interactive, perform and visualize impact analysis. It should facilitate and enhance the quality of communication among team members and help share knowledge.
R6	Empirical evidence	There should be sufficient empirical evidence regarding the practical validity of the estimation method, i.e., the estimation method has been applied and validated in a real-life setting.
R7	Data requirement	The estimation method should require a minimum amount of data (e.g., identification of most relevant class/es that could potentially be impacted due to implementation of a user story, a measure of relevant impact factors). The method should be able to use the data that already exists in a certain project context.

Requirements: An improved estimation method is required to have all necessary useful (understandable and complete) information. To keep and reuse this useful information, a historical database is also required that keeps not only the effort estimation trends but also impact data. This impact related information can help identify the number of system classes/ components affected by the user story or the number of test cases required, for retesting the system after implementing the user story. The method should be easy to understand and apply. It should require a minimum amount of data to generate useful information. If such a method is tool-supported, it should be interactive, perform and visualize impact analysis. Moreover, the tool should be easy to use. The introduction and application of the tool must not become an additional burden to the experts (developers). This would also facilitate communication among the team members.

3. P3: Subjective estimation does not support systematic learning.

Requirements: An improved estimation method is required to let experts become aware of the effect of impact factors. Firstly, the method is required to have understandable and complete information (impact and effort related). As mentioned earlier, such data would require a historical database to systematically store estimates together with the implemented change and contextual information for learning and reuse purposes. If such a method is tool-supported, it should support experts to compare estimates from completed projects and help with making new estimates. It should document effort data (estimated and actuals) and recognize patterns in estimates (for example, underestimation). This would let experts share knowledge. Furthermore, proper integration of tool and data, as well as their maintenance, should be ensured.

Table 2.6: Mapping of requirements to problems and goals

Requirements	P1-Low reliability of estimation output	P2-Limited informative power of the estimation method	P3-Subjective estimation does not support systematic learning	G1-To increase the estimation reliability	G2-To increase the informative power of the estimation method	G3-To improve learnability
R1. Reliability	x			x		
R2. Complexity	x	x		x	x	
R3. Informative power	x	x		x	x	
R4. Systematic learning support		x	x		x	x
R5. Tool support		x	x		x	x
R6. Empirical evidence	x	x	x	x	x	x
R7. Data requirement		x			x	

2.7 Summary

Based on published literature, and results of state-of-the-practice, it is recognized that expert-based estimation methods in agile development context are unreliable, lack useful information and do not support systematic learning. As a result of these problems, inaccurate estimates are produced that further lead to schedule and budget overruns. Experts remain unaware of the factors that influence estimation, and as they are incapable of making informed decisions, make biased estimates. To improve prevailing expert-based estimation methods, some requirements are also elicited. For example, if data on estimates are stored in a historical database, it could be re-used for estimating changes in the future. Additionally, explicit provision and consideration of useful information like the complexity and impact of changes on the underlying system would help experts take an informed decision as this information affects the magnitude as well as the accuracy of estimation. Furthermore, certain aspects of the estimation process, such as the potential impact of a change on the underlying system, if are tool-supported, can improve estimation effectiveness.

3 State-of-the-art

3.1 Introduction

This chapter describes the literature reviews of a) existing effort estimation (EE) methods in agile development method b) existing change impact analysis (IA) techniques proposed in the published literature. The identified EE techniques in agile development were evaluated to find the extent to which they fulfill the industrial requirements. Based on an analysis of the impact analysis techniques, the techniques that could be integrated into expert-based estimation and would provide with necessary impact information to support experts during estimation were identified.

This chapter is structured as follows: Section 3.2 describes the design, execution, and results of the review conducted for identifying existing EE methods in an agile development context. Section 3.3 gives an account of the evaluation of identified EE methods in an agile development context. Section 3.4 describes the design, execution, and results of the review conducted for identifying change impact analysis techniques. The chapter ends with a summary in Section 3.5.

3.2 Review of effort estimation in agile software development

3.2.1 Background

Various attempts, including data-based estimation models, have been made to improve EE in agile development context. However, it continues to be a challenge in the software industry [23, 113]. A secondary study, systematic literature review (SLR) that was published in 2014 [113] investigated works from 2001 to December 2013, resulting in a complete state-of-the-art analysis on EE in agile software development. In the other work done by the same authors [114], they presented results of the state-of-the-practice.

Since the review in the context of the thesis was made in 2015, therefore, there was a need to update the previous review. Therefore, by performing the forward snowballing technique [48] following the guidelines by Wohlin et al. [117], instead of automated search [21], a review was conducted to execute this update. The purpose was to find any other methods, tools or techniques that have been proposed for improving EE in an agile development context.

3.2.2 Review design and process

Since the review was an attempt to update the identified systematic literature review (SLR), the same research questions were investigated, and the same inclusion/ exclusion criteria were used in the evaluations as were in the referenced SLR [113]. Following were the research questions:

- RQ1: What techniques have been used for effort or size estimation in agile software development?
- RQ1a: What metrics have been used to measure estimation accuracy of these techniques?
- RQ1b: What accuracy level has been achieved by these techniques?
- RQ2: What effort predictors (size metrics, cost drivers) have been used in studies on effort estimation for agile software development?
- RQ3: What are the characteristics of the dataset/ knowledge used in studies on the size or effort estimation for agile software development?
- RQ4: Which agile methods have been investigated in studies on size or effort estimation?
- RQ4a: Which development activities (e.g., coding, design) have been investigated?
- RQ4b: Which planning levels (release, iteration, current day) have been investigated?

The review process following the forward snowballing [48] is as follows:

- The first step of forward-snowballing involves the identification of seed set i.e. a set of studies as a starting point. Since this was an updated review, therefore, the resulting studies found by the referenced SLR were considered as the seed set. Forward snowballing was performed on this seed set as described in the next steps.
- In the second step, search engines like Google Scholar [14] and Scopus [15] were used to analyze all the citations of all the papers in the seed set from 2014 to 2015.
- In the third step, the cited papers were subjected to the study selection procedure [19, 90]. Only peer-reviewed papers were selected. A basic evaluation was performed by analyzing the paper's title and abstract. An advanced evaluation was performed on the papers that passed the basic evaluation where every paper was read using the same inclusion/ exclusion criteria as described in the referenced SLR:

As general inclusion criteria, include all studies that:

- Are reported in English AND
- Are reported in a peer-reviewed workshop or conference or journal AND
- Are evidence-based (empirical studies)

If a study did not meet any one of the above conditions, it was excluded.

The remaining studies had to meet the following inclusion criteria:

- Report effort or size estimation related (technique or model or metric or measurement or predictors) AND
- Are based on any of the agile software development methods

A study meeting any of the following exclusion criteria was excluded from this review:

- Are not about effort or size estimation OR
 - Are not conducted using any of the agile software development methods OR
 - Deal with software maintenance phase only OR
 - Deal with performance measurement only, i.e., velocity measurement.
- In the fourth step, the studies that passed through the advanced evaluation criteria were selected and assessed for their quality using the quality assessment criteria of the referenced SLR.
 - In the fifth step, the data was extracted from the selected studies with respect to the research questions in the referenced SLR.

Snowballing is an iterative process. At the end of an evaluation cycle, a new cycle starts using the resulting papers of the previous cycle. The process ends when no more new citations are found. In this review, snowballing ended after one cycle. In addition to the 20 relevant papers found by the referenced SLR, we could find six relevant papers (see results in 3.1). This left us with a total of 26 papers (20 from the referenced SLR and six from snowballing) from which we extracted data and reported the results.

3.2.3 Results

This section briefly describes the results of this study.

Table 3.1: Papers searched through snowballing

Status	No. of papers
Total no. of papers	58
After evaluation	12
After quality assessment	6
Papers after snowballing cycle 1	6

- RQ1 - Estimation techniques: Planning Poker, in particular, was the most cited estimation technique, followed by ad hoc expert judgment. Contrary to referenced SLR, no Use Case Points (UCP) method was reported in this review. However, a trend towards the use of data-driven estimation models like neural networks was observed for estimation or supporting expert-based estimation methods.
- RQ1a - Accuracy metrics: Like the referenced SLR, Mean Magnitude of Relative Error (MMRE) is the most frequently used accuracy measure followed by Pred (n), Mean Square Error (MSE).
- RQ1b - Accuracy level achieved: Like the referenced SLR, the additional papers also did not report a good level of accuracy achieved. In most cases, it did not turn out to meet the 25 percent threshold. However, the studies that used data-driven estimation models presented somewhat better results.
- RQ2 - Effort predictors (size metrics, cost drivers): The most reported size metric was story points. Very few studies also used function points, most of the papers did not report the size metric. However, unlike referenced SLR, no UCP were reported. Among cost drivers, few studies report project and people related drivers like task complexity, priority, experience, technical ability, and team composition.
- RQ3 - Dataset used: Usage of both industrial and academic dataset was observed, however, use of industrial dataset and within the company data was dominant.
- RQ4 - Investigated agile methods: Few papers mentioned Scrum as the used agile method. Most of the papers do not mention any agile method.
- RQ4a - Investigated development activities: Most of the papers do not mention any development activity. However, few mentioned implementation as a development activity.
- RQ4b - Investigate planning levels: Most of the papers do not mention any planning level. Few papers mentioned the release planning level.

Since with the results of the updated review, only six relevant papers were found, the achievement of similar results in comparison to the referenced SLR did not come as a surprise. A slight trend towards using data-driven estimation models was observed nonetheless.

Satapathy [101] used a story-point-based approach and considered different machine learning techniques such as decision trees, stochastic gradient boosting, and random forests to assess estimation prediction. The results showed that the stochastic gradient boosting technique outperformed the others. In another study, Satapathy [100] used a story-point-based approach and considered different neural networks like General Regression Neural Network (GRNN), Probabilistic Neural Network (PNN), Group Method of Data Handling (GMDH), Polynomial Neural Network and Cascade-Correlation Neural Network. The results showed that the cascade network outperformed the other techniques used. Again in a different study Satapathy et al. [102] proposed various support vector regression (SVR) kernel methods. Optimizing the results obtained from a story-point approach, the purpose was to estimate the effort of agile software project more accurately. Performance comparison of these methods was made by using the data set from Zia et al. [120]. Based on the values of MMRE and Pred, it was found that radial basis function (RBF) kernel-based SVR outperformed other proposed kernel methods. Raslan et al. [94] also proposed a fuzzy-logic-based EE framework considering user stories. Popli et al. [92] considered story-point-based regression analysis for estimating effort but did not describe any accuracy metrics. They did consider certain people, project and resistance factors in devising their proposed model.

A lack of empirical evidence on the applicability and effectiveness of these models in real life setting is observed as some of them validated their approaches only in academia.

Overall, the state-of-the-art analysis on EE in an agile context, as captured by the referenced SLR and our updated review, showed expert-based estimation methods (i.e., expert judgment, planning poker) as the most dominant methods followed by data-driven estimation models. Furthermore, some studies have used function points. However, function points are not commonly used in the industry. Parvez [89] modified the use case point estimation method by adding efficiency and risk factors to estimate testing the effort in of agile projects. The approach showed improved results for estimating the testing effort. Similarly, Husain et al. [58] proposed a methodology that finds COSMIC functional size from informally written textual requirements for estimating development effort in an agile development context. Kang et. al [65] proposed a dynamic software cost estimation model (using function points) and a project tracker (using a Kalman Filter) on daily-based velocity.

Nonetheless, story points remained the most used size metric which is more relevant for the agile context. Zia et al. [120] developed a story-

point-based regression model. It uses a ranking mechanism to specify story size and complexity, considers and ranks certain other factors that impede the velocity of creating a model and used 21 projects to calibrate it.

Bhalerao et al. [32] proposed a Constructive Agile Estimation Algorithm (CAEA) which incorporates factors believed to be vital for accurately investigating the cost, size, and duration of a project. Moser et al. [84] proposed a model using neural networks and code metrics to predict the development effort of a user story. This approach was extended by Abrahamson et al. [17] who extracted and used keywords from user stories instead of code metrics and then built models (regression, neural networks, support vector machines) to predict the development effort of the user story. The results show that EE works well only if the developers write well-structured user stories.

None of the estimation methods (expert judgment) in agile development have leveraged the tools and techniques that can perform change IA to estimate the effort required to implement a change and takes care of its impact also.

3.3 Evaluation of existing effort estimation methods

This section presents the evaluation of the estimation methods identified through the state-of-the-art analysis in the context of agile software development. These methods include all the methods identified through the referenced SLR as well as the resulting methods from our updated review. For the sake of simplicity, the methods were grouped into categories depending on the type of estimation method used, i.e., all the studies that mentioned used planning poker were grouped under Planning Poker. The corresponding references are also provided. The studies that did not provide any specific technique but discussed other related topics were not included in the evaluation. For example, Cao [43] showed in general that the estimation in agile development is more accurate than traditional development and that accuracy has not improved in agile development over time. Similarly, in [81] effects of certain factors on estimation was found. It was shown that projects where collaboration was facilitated by daily communication between the contractor and the customer, exhibit a lesser magnitude of effort overruns. In [70] influence of social factors on estimation was analyzed. Such studies were excluded from evaluation as they could not be categorized and also none of the criteria was applicable to them.

The methods were evaluated using criteria (C1 to C8) defined in Table 3.2. We have adapted and used the criteria, scale and the threats to validity as defined by [111]. These criteria were derived from the requirements regarding estimation methods listed in Table 2.5. A four-point Likert scale was used to evaluate each criterion. Using this scale,

a given estimation method was evaluated by rating the extent to which each method supports a certain requirement. The scale was used with the symbols as follows: Strongly supports (++), moderately supports (+), weakly supports (-), no support (--). The evaluation is presented in Table 3.3. The evaluation is based on the author's experience as well as on the data extracted from the identified estimation methods. The threats to the reliability of the evaluation in Table 3.3 may be due to the evaluation metric. The threats include 1) subjective definitions of the criteria, 2) Fuzzy definition of the metric and 3) Human judgment error.

From the evaluation of existing estimation method in Table 3.3, very few estimation methods were found to be reliable. We have found studies that mentioned an accuracy metric for their applied estimation method but did not report the results. For such studies, we marked a single "-" in "Reliability". Moreover, certain studies have applied and compared several estimation techniques, out of which only one seemed to perform better in terms of accuracy. For such studies, we marked a single "-" in their "Reliability". For example, in the study by Satapathy et al.[101] where various SVR Kernel methods were applied and compared, only the RBF Kernel was found reliable. Same is true for their other study [100], in which various neural networks were applied however only cascade networks were found reliable.

Methods comprising of data-driven estimation models, e.g., GRNN [100], SVR/RBF Kernel methods [101] were found too complex in their underlying theory.

None of the methods in Table 3.3 possessed "Informative power", or were able to provide "Learning support". For the estimation techniques where certain criteria were not applicable we marked "NA" in the corresponding cell, e.g., for expert judgment and Planning poker, we marked "NA" for "Tool support".

Most of the methods required a lot of data as input e.g. UCP [89], CAEA [32], SVR/RBF Kernel methods [101].

A single "+" in "Empirical Evidence" shows a lack of information on the used data set for example, Tamrakar et al. [104] and Mahnic et.al [76] did not provide information in used data set. Similarly, there were studies from Bhalerao et al. [32], Catal et al. [44] and Logue et al.[75] that have applied an estimation technique but did not measure their predicted accuracy. Such studies may be of negligible use to both researchers and practitioners as they did not provide any solid evidence on how accurate the proposed effort estimation techniques were. For such studies, we marked a single "-" in "Empirical evidence".

Despite this research on improving EE in an agile context, a lack of empirical evidence on the accuracy of these methods (or models), a lack of information on the used data set to build models and/or size metrics and/or cost drivers is reported by the review [113]. Moreover, most of

Table 3.2: Estimation method evaluation criteria

Criterion	Definition
C1. Reliability	The reliability of the method is measured in terms of accuracy of estimation output. The extent to which the estimation method produces accurate outputs. The accuracy of method can be measured using existing performance metrics e.g. Mean Absolute Error (MAE) which is the average of the absolute errors between the actual and the predicted effort, Mean magnitude of relative error (MMRE), Prediction Accuracy PRED (x) which is the average of MAE's off less than or equal to x. The accuracy of the estimates directly corresponds to PRED(x).
C2. Complexity	The extent of the complexity of the underlying theory/technique employed in the estimation method.
C3. Informative power	The extent to which the estimation method provides additional useful (understandable, complete) information that contributes to the achievement of estimation objectives, e.g.(support experts to make informed decisions while doing estimation, support systematic learning, facilitate communication and knowledge sharing). This information includes relevant code-based factors that have an impact on effort, historical effort related data.
C3.1 Understandability	The extent to which the information provided is in an understandable way.
C3.2 Completeness	The extent to which the information provided is complete.
C4. Systematic learning support	The extent to which the experts performing effort estimations are aware of or learn about the factors influencing estimations.
C5. Tool support	It refers to the level and quality of tool support. It includes but not limited to quality aspects like ease of use, usability, interactivity, visualization.
C6. Empirical evidence	The extent to which the method has been evaluated in practice. This includes considering the context (academic or industrial), and the reliability of documented in-field applications of the method.
C7. Data requirement	The amount and type of data required by the estimation method. The amount refers to the number of projects or factors (e.g., COCOMO II [1] model requires 17 effort multipliers and 5 scale factors). The data type refers to the measurement scale (nominal, ordinal, etc.)

the improvement approaches either improve expert estimation by providing additional factors [32] to consider during estimation or by proposing a data-driven static model [100] that keeps expert out of the estimation process just like the estimation models proposed for traditional development context thereby violating the Agile Manifesto [30].

Table 3.3: Estimation methods evaluation

Estimation method	C1. Reliability	C2. Complexity	C3. Informative power	C4. Learning support	C5. Tool support	C6 Empirical evidence	C7. Data requirement	Reference
Planning poker	--	--	--	--	NA	-	-	[55][104][98][71][72]
Expert judgment	--	--	--	--	NA	--	-	[17][75][89][97][55][44]
Use Case Points (UCP) Method, UCP Method Modification	+	-	--	--	--	+	++	[89]
Linear regression, Neural Nets (SVM)	+	++	--	--	--	+	+	[17]
Robust regression, Neural Nets (RBF)	+	++	--	--	--	+	+	[16]
CAEA	-	+	--	--	--	--	++	[32]
SVR: Linear Kernel, Polynomial Kernel, RBF Kernel, Sigmoid Kernel	-	++	--	--	--	+	++	[101]
GRNN, PNN, GMDH, Cascade-Correlation Neural Network	-	++	--	--	--	+	++	[100]
Kalman filter	-	++	--	--	--	+	+	[65]
Cosmic FP	-	+	--	--	--	+	-	[58]
Statistical combination of individual estimates	+	-	--	--	--	+	-	[76]
Own algorithm	-	-	--	--	--	--	++	[92]

3.4 Review of change impact analysis

3.4.1 Background

Several studies on change IA techniques exist in the literature. Therefore, a systematic search was conducted to identify any secondary studies on the topic.

As a result, a secondary study, a survey on code-based change IA techniques by Li. et al. [73] was found. It was published in 2013 and investigated works from 1997 to 2010, resulted in a complete state-of-the-art analysis on code-based change IA techniques.

Since the review in the context of the thesis was made in 2015, therefore, there was a need to update the previous review. Therefore, by performing forward snowballing technique [48] following the guide-

lines by Wohlin et. al [117] another review was conducted to execute this update.

The purpose was to find a code-based change IA technique that meets some of the criteria (as mentioned in 3.2), i.e., it is simple, understandable, requires less amount of data and effectively provide all impact related information (i.e., the factors as discussed in Table 2.2). Moreover, it can be integrated with EE technique to provide experts with impact related information and support them in estimation.

3.4.2 Review design and process

As the purpose was to find a suitable change IA technique, a subset of the same research questions was investigated, and the same inclusion/exclusion criteria were used in the evaluations as were in the referenced survey [73]. Following were the research questions:

- RQ1: What techniques/approaches have been used to perform the change IA?
- RQ2: Which properties can be identified to characterize the research on code-based change IA?

The same review process followed was the same as described in Section 3.2. The only difference was the selection of the seed set to start the review. In Section 3.2, the seed set were all the articles selected by the referenced SLR. Here the seed set was this secondary survey only. It is firstly because the purpose was to find an IA technique that could be utilized in the context of this thesis. Secondly, since the review represented the complete state-of-the-art on code-based change IA techniques we, therefore, restricted the seed set to this review.

In addition to meeting the general inclusion/exclusion criteria in Section 3.2 the inclusion criteria for this review aimed at including the studies that:

- Focused on a specific change IA technique, stated the change IA to be one of its goals and provided empirical validation of the technique from the change IA perspective.

The exclusion criteria for this review aimed at excluding the studies that:

- Focused on some other problems rather than the change IA technique OR
- Focused on the traceability-based analysis techniques or high-level model-based (design level and requirement level change IA) rather than the code-based change IA technique.

In this review, snowballing ended after one cycle. In addition to the 23 unique IA techniques found by the referenced review, we found only five relevant papers (see results in 3.4). However, none of these five papers had any new or unique IA technique that was not mentioned in the referenced review, therefore, we did not consider them.

Table 3.4: Papers searched through snowballing

Status	No. of papers
Total no. of papers	43
After evaluation	7
After quality assessment	5
Papers after snowballing cycle 1	5

3.4.3 Results

This section briefly describes the results of this review.

- RQ1- Techniques used to perform IA:
The result of this analysis indicated that two main types of analysis had been used in the published literature: static analysis and dynamic analysis each with subtypes. Static change IA techniques take into account all possible behavior and inputs of the software. They are usually performed by analyzing the syntax and semantic, or evolutionary dependence (i.e., change history repositories). Subtypes of static analysis are as follows:
 1. Structural static analysis. It focuses on static analysis of the structural dependence of the program and construction of the dependence graph. Knowing the structural dependence allows predicting which elements are impacted based on this dependence.

In these techniques, the impact set is found by performing a reachability analysis on the dependency graph. For example, Badri et al. [28] and Hattori et al. [53] used the control call graph to perform static change IA. Huang et al. [57] performed dependency analysis in object-oriented programs. Petrenko et al. [91] made use of a hierarchical model to obtain the impact set. It is found that some change IA techniques operate using coupling measurement like structural coupling. For example, Briand et al. [39] used object-oriented coupling measurement to identify the impact set.
 2. Textual analysis: It extracts conceptual dependence (conceptual coupling) based on the analysis of the comments and/or identifiers in the source code. Torchiano et al. [110] pro-

posed using source code comments and change logs in a software repository. Some change IA techniques operate using coupling measurement like conceptual and relational topical based coupling. For example, IA proposed by Poshyvanyk et al. [93] is based on conceptual coupling. It was found that one of such couplings (i.e., the maximum conceptual coupling between two classes), was better than existing structural coupling measures when compared in terms of the accuracy of impact set. Gethers et al. [50] used relational topic based coupling to capture topics in classes and relationships among them for change IA. They also showed that this coupling complements the conceptual coupling proposed by Poshyvanyk et al. [93].

3. **Historical analysis:** It is performed by mining information from multiple evolutionary versions of the software in software repositories. IA techniques using historical analysis are based on identifying the co-change coupling between the entities that are changed together. The co-change coupling information helps predict future changes. In this direction, Zimmermann et al. [121] applied data mining to version history repositories to extract the co-change coupling between at the file level. In another technique, Hattori et al. [53] incorporated two data mining algorithms like Apriori and DAR in the repositories for change IA.

Historical analysis is found to be the most used technique followed by structural static analysis. However, both of these types cannot identify conceptual dependence. For example, the identifiers and comments of the source code made by the developers reflect the implicit relationships (dependencies) among parts of the same software. Analyzing these relationships is also important as there is a likelihood that they refer to similar concepts in the problem or solution domains of the software. Hence, textual analysis is needed to cover these dependencies hidden in the comments and identifiers.

Techniques have also been proposed that combine different types of change IA. For example, Kagdi et al. [64] used a combination of textual analysis and historical analysis. They utilized both source code of the current program version and previous versions from software repositories to obtain better impact results when compared with using them independently. Canfora et al. [42] also combined textual and historical analysis. They used textual similarity to retrieve past change request in the software repositories for change IA. Huang et al. [57] used a combination of structural static (traditional dependency analysis) and dynamic analysis in object-oriented programs for performing change IA.

Dynamic IA involves considering specific inputs like test data and relies on the analysis of the information collected during program execution (e.g. execution traces information, coverage information, and execution relation information) to estimate the impact set. Subtypes of dynamic IA include offline and online analysis.

Online change IA is performed by using the collected information while the program is executing whereas offline IA is performed as soon as the program finishes its execution. The purpose of the online change IA was to alleviate the need to obtain the whole runtime execution information when only part of it is required.

Technique by Beszedes et al. [31] used dynamic function coupling between two functions for change IA. Law et al. [67] provided a technique for dynamic change IA based on whole path profiling whereas Breech et al. [37] proposed an approach of the whole program path-based dynamic online impact analysis. Apiwattanapong et al. [26] proposed the execute-after relation between entities to support dynamic change IA. The precision of online and offline impact analysis has also been empirically validated [36][37]. The results show that online change IA techniques compute equally precise impact set as offline techniques. However, they scale better [36].

Dynamic IA is more expensive than static IA due to the overhead of expensive dependency analysis during program execution. Moreover, the impact set identified through dynamic IA often includes false-negatives.

- RQ2- Categorization of code-based change IA:
Li et.al [73] identified set of properties, and Lehnert [69] derive a taxonomy from categorizing the IA techniques. We extended Lehnert's taxonomy by combining it with the results of Li's survey. To facilitate the selection of appropriate IA techniques by practitioners, we used the following facets and sub-facets in this extended taxonomy to categorize existing IA techniques:
 - Context: change request metadata, type of change, development phase, available artifacts.
 - Impact analysis: IA technique used, type of analysis, change impact type, the granularity of impact set, the priority of impact set, tool support, supported language, application area.
 - Empirical evidence: benchmark size, performance measures used, results, scalability.

In general, most of existing change IA techniques can be used in object-oriented programmes, and there is a need to develop techniques for other programming paradigms, for example, aspect-oriented programmes.

Applications of impact analysis include an understanding of the system with respect to a certain change, implementing the change and retesting the modified parts [24, 79]. JRipples [41] supports program comprehension and change propagation during incremental changes. It analyzes the program and automatically highlights the impacted classes as the developer implements a change.

One needs to be aware of the possible ripple-effects caused by implementing a new change in the existing system, so nothing is overlooked. JTracker [52] assists in change propagation. Whenever a developer changes a class, JTracker marks the potentially impacted neighboring classes. Then retesting the newly implemented change through regression tests and identify other test cases that may need to be done for maintaining the existing system intact and error free [22, 35]. Chianti [95] can be used for regression testing. It selects the test cases that need to be rerun to test the implemented change and also adding new test cases to cover the affected parts not tested by the original tests.

It is found that with the capability to identify software life cycle objects that are likely to change for each change request, impact analysis can support project managers to determine the effort and cost estimates for software change more precisely [35]. Furthermore, these estimates support software release planning by providing important information like what can or cannot be included in a release over a given time period. Impala [54] supports IA before the change is implemented. It searches in the dependency graph, the elements of the change set depending on change type and dependency type, returning reachable elements in the change set. ImpactMiner [46] estimates an impact set from natural language change request query using an adaptive combination of static textual analysis, dynamic execution tracing, and mining software repositories techniques [50].

Despite the research on change IA techniques, only a few research prototypes have been implemented that support some application of change IA in traditional software development. Commercial tools for applying IA are sparse. The research prototypes mentioned above are also not actively maintained.

It has become evident from the state-of-the-art of estimation methods in Section 3.2, none of the methods so far have used change IA information for planning purposes. Therefore, further research is required regarding which change IA technique can be utilized in supporting estimation and planning activities in addition to finding which technique to select and how.

3.5 Summary

The review of EE methods in agile development context revealed that expert-based estimation methods (i.e., expert judgment, planning poker) as the most dominant methods. A trend towards the use of data-driven estimation models was also observed. Expert judgment remained inaccurate whereas a lack of empirical evidence on the applicability and effectiveness of the data-driven models in real life setting is also observed nonetheless. An evaluation of these methods showed several deficiencies towards fulfilling industrial requirements.

The review of change IA methods revealed that a large number of techniques exist in published literature using various change IA types. Historical analysis is found to be the most used technique followed by structural static analysis. A combination of IA types computes more accurate impact set as compared to using them independently. Most of these techniques were empirically validated using several measures including accuracy of the estimated impact set.

Even though IA is a well-researched field, commercial tools for applying IA are sparse. The research prototypes mentioned above are also not actively maintained. Due to these constraints, it was not possible to use any existing tool for utilizing IA in the context of this thesis.

It is also established that none of these methods have leveraged the tools and techniques that can perform change IA to estimate the effort required to implement a change. It is because impact analysis is not well integrated into current estimation methods. This thesis, therefore, aims to fill this gap by developing a systematic hybrid effort estimation method that increases estimation reliability, enables learnability and increases the informative power of expert-based estimation. It does so by utilizing change IA techniques and integrating them in effort estimation techniques.

4 HyEEASe - Method for effort estimation in agile software development

4.1 Introduction

This chapter introduces the development of a proposed effort estimation method - HyEEASe. HyEEASe is a hybrid, lightweight and systematic method for estimating the effort in the context of iterative and incremental software development. The method is a result of synthesizing research on effort estimation (EE) and change impact analysis (IA) [73]. It utilizes IA information for supporting a human judgment based estimation [106]. Static code analysis and historical change impact analysis are used to isolate the potentially impacted parts for a change. Furthermore, past estimation and data related to the impacted parts are also presented to practitioners when they are estimating effort for a new change. Additionally, a Gradient Boosted Trees (GBT) based estimation model is also developed.

To develop and implement the proposed estimation method several empirical methods were employed. Through case study and survey research, the needs of industry practitioners, and their challenges and development context were understood. The input from these studies and the results of systematic literature reviews were used to develop the method incrementally. Initially, it was developed in close collaboration with SAP and the potential tool-support was materialized using mock-ups. Encouraged by the results, using prototyping the method was further refined in close collaboration with Insiders Technologies.

In this chapter, first, a conceptual framework to facilitate the process of finding a change IA technique and then integrating with an EE technique is described. Then, the IA and EE techniques that were selected and integrated in HyEEASe, the high-level design of HyEEASe, its workflow and development are described. The research done for this chapter was conducted as part of research projects, Abakus, grant number 01IS15050G, and HyEEASe, grant number 01IS12053 funded by the German Ministry of Education and Research (BMBF). The results were also published, details can be found at [105], [107] and [109].

This chapter is structured as follows: Section 4.2 gives insights into the proposed conceptual framework for the selection and integration of IA and EE techniques. Section 4.3 describes the selected IA and EE techniques in the context of this thesis. Section 4.4 illustrates high level design where as Section 4.7 details the work flow. Section 4.5 explains the

development of the impact analysis model that was used in HyEEASe. In Section 4.6 the development of an estimation model is explained. Section 4.9 discusses the limitations of HyEEASe. The chapter concludes with a summary in Section 4.10.

4.2 Conceptual framework

The need for a framework to guide the use of change IA for improving EE was identified to leverage the support provided by change IA techniques and tools. Therefore, a conceptual framework [107] on how to select an IA technique and guidelines [105] on how to operationalize this framework has been proposed. The framework is inspired by the Quality Improvement Paradigm (QIP) [29]. The framework is expected to support researchers in selecting appropriate IA and EE techniques in a given organizational context. Additionally, it will support practitioners in improving their estimates. By reusing the knowledge and experience gathered in each iteration, the framework enables continuous improvement towards effective EE in practice. The overall framework and the process of integration are depicted in Figure 4.1 and Figure 4.2 respectively. The six steps of applying the framework in the given context (agile development) are as follows:

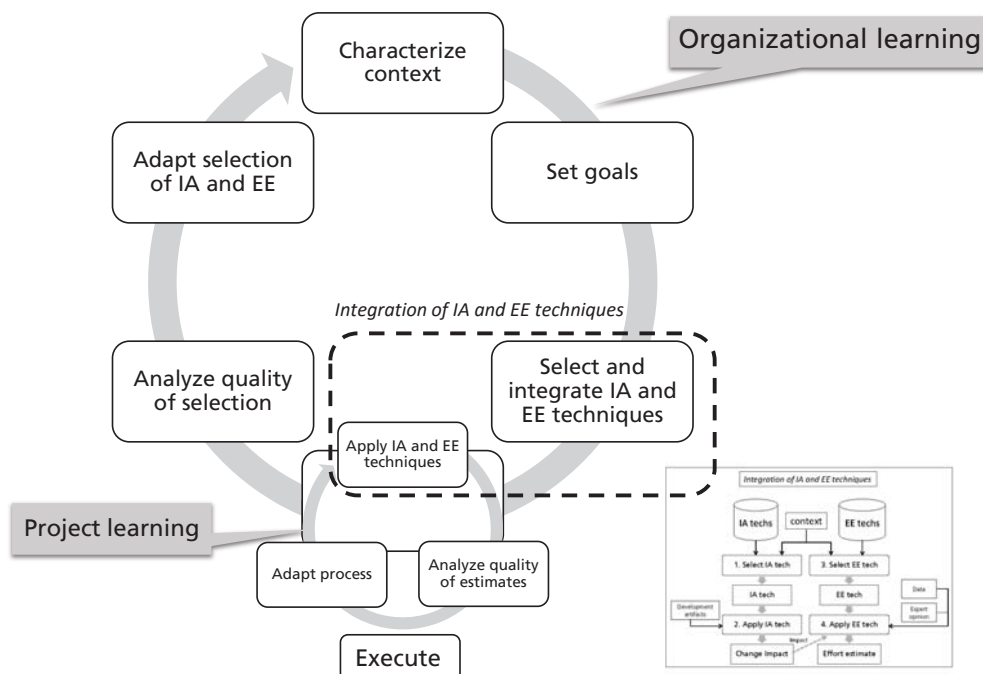


Figure 4.1: Framework for integrating impact analysis with effort estimation for improving estimation

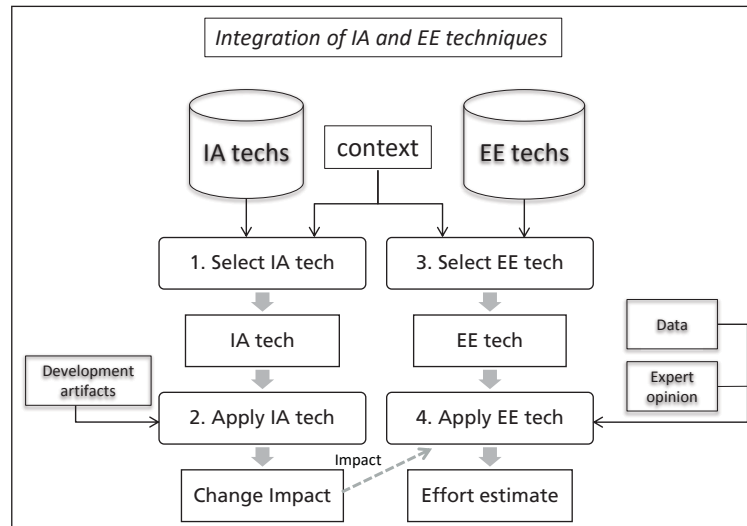


Figure 4.2: Integration of IA with EE techniques

1. **Characterize context:** In this step, the organizational and project context concerning EE and IA technique is specified. For example, the practitioners specify the estimation purpose, development process (e.g. Scrum, XP [27], DevOps [59, 60]), level of planning, prevailing estimation method, personal factors (experience, knowledge), estimation trend, estimation error, available development artifacts and their level of granularity (e.g., requirements/ user story/ BLI), type of change, product factors (product complexity, used technologies, tool support, requirements on tools) etc. This information is stored in an experience base to be reused later on in sprints.
2. **Set goals:** In this step, the practitioners specify a measurable/quantifiable and concrete improvement goal for the estimation process. For example, the goal may be to reduce the error margin in effort estimates by 20%.
3. **Select and integrate IA and EE techniques:**

The selection of both IA and EE techniques is driven by the organizational context. It is that observed a two-way relation exists while selecting an IA technique for utilizing in EE, let's call it as a "required-provided by" relation, i.e., what is required by the IA technique (as input) and what is provided by the organization, and vice versa. The extended taxonomy mentioned in the previous section can be explored additionally based on this relation to decide on the selection of an IA technique. The process of selection, application, and integration of IA and EE techniques is shown in Figure 4.2. In step 1 and step 3 of Figure 4.2 taken from a collection of IA and EE techniques, the practitioners can select appropriate

IA and EE techniques, respectively, based on their context, using proposed guidelines [105].

4. Execute: The process of applying and integrating IA and EE techniques is executed as follows:

- (a) Apply and integrate IA and EE techniques: In this step, first, the selected IA technique is applied to the given software to find the impact and then it is integrated with the EE technique, using this impact as input to EE, to find the estimates.

The initial estimate produced as a result of this integration is further refined reusing previous knowledge and data gathered during the sprints.

- (b) Analyze quality of estimates: The estimates made are assessed by the practitioners at the end of a sprint in the retrospective meeting or estimation workshop. The estimated values are extracted from the experience base and compared to the actual ones. In case of deviations, further investigation is required to find the cause of the deviation and feedback is provided to minimize the estimation error. To do this analysis, other data such as historical effort data, data regarding change requests, impact data, and effort overhead (factors) are required in addition to expert opinion and are retrieved from the experience base.

- (c) Adapt process: In this step, the feedback gathered in 4b is considered to adapt the process for the next sprints. To adapt the integration process, effort factors (as also identified by the case study [108]), previous knowledge of the practitioner, and the data gathered in 4b are considered. Also, any further requirements regarding the guidelines for integration are elicited from the practitioners.

Using the parameterization of the estimation process based on factors and the feedback of the expert, the estimation process is adapted for the next sprint. This knowledge is also stored in the experience base and is reused in the next sprint. Through this small circle, practitioners learn about their project and estimation performance over sprints.

5. Analyze quality of selected techniques: At the end of a project release, the selected IA and EE techniques are analyzed regarding their suitability and effectiveness for the estimation process and whether they could achieve the specified goals. Information like the feedback of the practitioners, stored estimation data, and knowledge, as well as the adaptations to the integration process over sprints, is required and is retrieved from the experience base to analyze the effectiveness of the selected IA and EE techniques.

6. **Adapt selected techniques:** In this step, if the selected techniques are found unsuitable or ineffective and the set goals are not achieved, practitioners discuss and explore other techniques from the collection of IA and EE techniques that might be suited better in a given context. Through this cycle, the organization learns about their projects over time.

In the given agile development context, organizational level learning refers to experience and knowledge gathered from release to release, whereas project level learning refers to knowledge acquired from sprint to sprint. The feedback can be provided in the context of retrospective meetings or by arranging additional estimation workshops - whichever is feasible for the organization.

The steps 3 and 4 are the major building blocks of the proposed framework. Based on the context, the practitioners can select and integrate IA and EE techniques using guidelines in these steps. The practitioners can then use IA while making estimates to improve them.

4.3 Selected impact analysis and effort estimation techniques

The selection and integration of IA and EE techniques were done through the use of the proposed conceptual framework described in Section 4.2. The results of the state-of-the-art analysis of effort estimation in agile development, indicate the expert-based estimation methods are most often used, with "Planning Poker" being the most frequently employed method [114]. This motivated the choice of "Planning Poker" as the estimation method that will be supported by IA techniques. By doing so, the scope of the thesis was reduced in terms of the selection of estimation techniques.

From the review of change impact analysis techniques, it was found that there exist two main types of analysis approaches, which are static and dynamic IA. Dynamic IA involves program execution and the collection of the execution trace information. It is, therefore, more expensive than static IA due to the overhead of expensive dependency analysis during program execution. Moreover, the impact set identified through dynamic IA often includes false-negatives. Due to these reasons, dynamic IA was not considered the optimal option for performing IA in this thesis.

Moreover, static IA has three types, i.e., structural static analysis, textual analysis, and historical analysis. Textual analysis extracts conceptual dependence (coupling) based on the analysis of the comments and/or identifiers in the source code. It relies heavily on the developer's writing skills and choice of words to describe implicit relationships in identifiers and comments of source code. Therefore, there is a risk that different developers used different vocabulary to express the same functionality or vice versa. This requires natural language processing for analyzing

code comments and identifiers which in itself is a research topic and is certainly out of the scope of the thesis.

In general, an impact set is required to start an IA, while the impact volume is expected as the output of an IA. The impact set is comprised of component/s identified by the experts after analyzing the change request (or backlog item (BLI) or user story). The volume of impact means, “what” the impact was (i.e., which component/s were changed in the past concerning a particular user story) and “how much” it was (i.e., quantified information about the impact).

In the context of this thesis, it was realized that historical analysis could be used to find out what (which classes) were changed, and structural static analysis could be used to find out how much the identified classes were changed. Since a combination of IA techniques can provide better results (impact) than the consideration of single techniques [73], therefore, structural static and historical IA were adapted and applied. With this selection, the scope of the thesis was also reduced in terms of IA techniques selection.

The quantification of the impact is specified by measuring the changes through structural static IA using certain code metrics. In Section 2.3 Table 2.2, code-based impact factors were identified. As these factors were considered very important for estimation, we quantified them using code metrics [45] as shown in Table 4.1.

Table 4.1: Code-based impact factors and their metrics

Code-based impact factor	Metric name	Explanation
BLI impact on parts of the developed system.	Size in LOC/SLOC	Refers to the number of lines that contain source code.
BLI complexity.	Cyclomatic Complexity or Weighted Methods per Class (WMC)	Refers to the complexity of the potentially impacted methods/classes measured as McCabe’s Cyclomatic Complexity. It is the sum of the cyclomatic complexity of all nested functions or methods).
Dependencies among BLI.	Coupling Between Objects (CBO)	Refers to the coupling of the potentially impacted methods/classes. CBO is a count of the number of other classes to which it is coupled. Class A is coupled to class B if class A uses a type, data, or member from class B.

The code metrics (LOC, CBO, WMC) (as described in Table 4.1) for the method level were chosen as these were found to be the most relevant ones in IA research as well as in the state-of-the-practice analysis for the task of effort estimation. Li et al. [74] took five metrics from Chidamber and Kemerer [45], added three of their own, and found a strong corre-

lation between these metrics and maintenance effort. Briand et al. [40] also found relationships between most of the coupling/cohesion metrics and fault-proneness of classes. They further examined if coupling measures could be used for estimating an impact set [39]. Antonio et al. [25] used the size of the modification, which is a key factor for maintenance effort and cost estimation, for estimating object-oriented programs. Similarly, Zimmermann et al. [121] applied data mining to version histories to extract the co-change coupling between files for performing IA.

For each identified class in the impact set, the quantified added and deleted values of code metrics (LOC, WMC, and CBO), as well as the changed code metrics of co-changed classes, are returned through static IA. Co-changed classes refer to those classes that were changed together in the version control system while implementing a user story for the same sprint. Such classes seem to have co-change coupling [121]. The information required for historical analysis comprised of the following:

- Class Name (class in the identified impact set).
- User story ID (previously implemented user story associated with the new user story via a similar class). "Similar" in the current context refers to the common classes impacted/used between new and already implemented user stories.
- User story type (defect or new feature).
- User story description, Actual effort, Estimated effort, Estimation trend (overestimated or underestimated or just right).
- Difference between actual and estimated effort.
- Total number of affected classes (for respective user story) as well as the details of co-changed classes.

Even though IA is a well-researched field, commercial tools for applying IA are sparse. What is most commonly found are research prototypes, which are often not actively maintained. Due to these constraints, it was not possible to use an existing tool for utilizing IA in the current context. Thus, a prototype tool was developed in MS Access 2010 to support the method.

HyEEASe integrates structural static and historical impact analysis techniques with the Planning Poker estimation technique. It is intended to support the experts in identifying the potential impact of a new user story (or BLI) based on the quantified impact information of a similar user story implemented in the past, i.e., based on what and how much was changed in the past.

4.4 Overview of the design

HyEEASe is composed of three main processes as shown in Figure 4.3 and explained below:

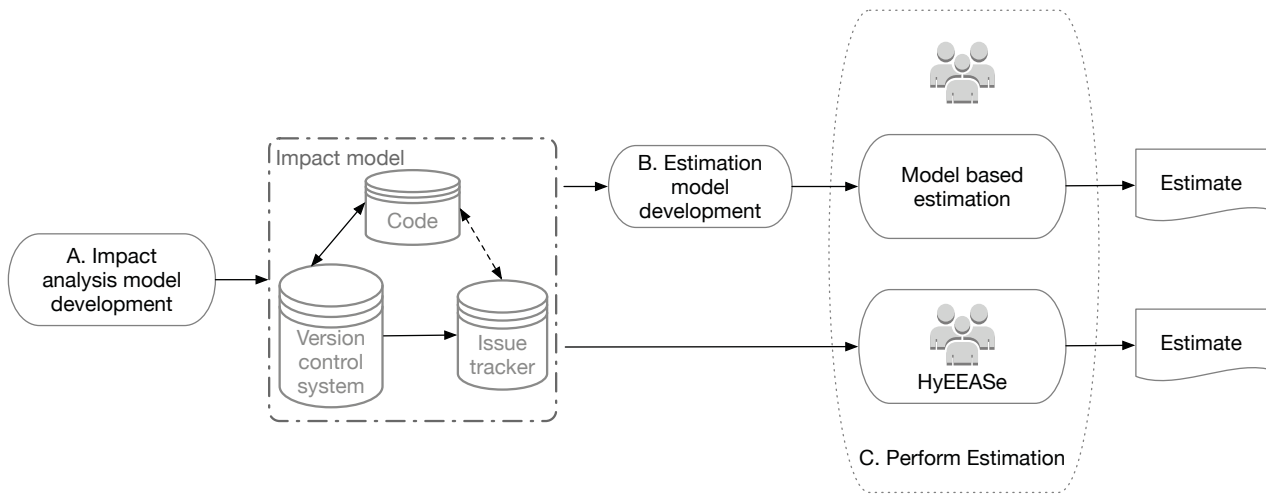


Figure 4.3: High-level design overview

- **A. Impact analysis model development:**
In this process, the impact analysis model is developed using static and historical IA.
- **B. Estimation model development:**
In this process, the GBT model is developed using a subset of data that is extracted through an impact model. The impact and the estimation model are developed only once for a project and are updated after each sprint.
- **C. Perform estimation:**
In this process, experts perform estimation for all user stories of a sprint using either of the two paths, i.e., Hybrid estimation or GBT model estimation as shown in Figure 4.3.

4.5 Impact analysis model development - Process A

To build the impact analysis model, three major components, i.e., a code repository, a version control system, and an issue tracker system were required. Generally, in software development companies, the user stories are stored in the issue tracking system with a unique ID. In impact model in Figure 4.3, the code is directly (straight line) connected to its version control via commit messages. These commit messages keep this ID of the user story when the code is checked into the repositories and are thus traceable. Therefore, the code is indirectly (dashed line) connected to the issue tracker via these user story ID tags in the impact model in Fig-

ure 4.3. To extract impact information effectively, certain pre-requisites, while committing changes to the version control, are as follows:

- The user story ID should be placed in the commit message while committing.
- Changes related to only one user story should be committed through one commit at a time (though there may be multiple commits for any user story during a sprint).

The impact analysis model comprised of static and historical impact analysis results. It is shown as a dashed rectangle A. Impact analysis model development in Figure 4.3. For performing static and historical impact analysis, the following data was required:

- User story metadata like ID, type, description, sprint date, commit ID and effort data.
- A mapping displaying all user stories that affect a certain class (*class* \rightarrow *userstory*).
- The total number of classes affected when a single user story is implemented.
- The quantified code metrics for each affected class, i.e., LOC, WMC, CBO.

The process of impact analysis model development is as follows:

1. A.1 Data collection:
Input: Issue tracking system, source code, version control system.
Process: The data was collected from three components, i.e., the issue tracking system, the version control system and the source code as shown in Figure 4.3. A brief account of implementation on how data collection was done for the impact model, is as follows:
 - A.1.1: The user story information was stored coherently in a database, which was extracted using SQL queries. There existed an $m \times n$ relationship between a user story and a class. A user story can affect multiple classes. However, capturing these affected classes was complicated because the information on which the user story affected which class/es is only stored implicitly in version control. The version control system stored the affected "files" instead of storing "classes" against each commit.
 - A.1.2: To gather the affected classes, first, every commit had to be captured with its respective user story and analyzed the affected files.
 - A.1.3: A script was created that mapped each user story to its corresponding commits.

- A.1.4: For each commit, the changed files were analyzed using a static code analyzer namely the Understand tool™ [12]. The output of this analyzer consisted of class and file metrics (including the affected class names), which was used in combination with the version control information to create the *class* → *userstory* mapping for each class.

Output: Mapping of user stories and affected classes to corresponding commits. Code metrics for each affected class at the commits.

2. A.2 Perform static impact analysis:

Input: Mapping of user stories and affected classes to corresponding commits. Code metrics for each affected class.

Process: The data were pre-processed for creating an impact analysis model with the information from static impact analysis. A brief account of the implementation of how data was prepared for the impact model, is as follows:

- A.2.1: Changes in LOC, WMC, CBO metrics in both dimensions (added/deleted) were collected at the commit level.
- A.2.2: For any deleted and any newly added files at a certain commit, data from all the commits in a sprint for a certain user story was synchronized. Classes with no changes were discarded.
- A.2.3: The information about the changed metrics for each affected class regarding each specific user story was then aggregated. This was done to determine how much was changed in an affected class for a specific user story in a given sprint. An example illustration is provided in Figure 4.4. It shows an excerpt from the output of A.1 and A.2, i.e., the commit (Cm_i), user story (US_j) and class (Cl_k) relationship. In the first table, it shows seven commits, each connected to a user story that further is related to the classes that were affected in terms of LOC. In the lower table, the illustration shows the aggregated commits for the same user story and the resulting change (LOC) each affected class in that user story.

Output: Impact analysis model with information obtained from static impact analysis.

3. A.3 Perform historical impact analysis:

Input: Mapping of user stories and affected classes to corresponding commits.

Process: Historical effort data for these user stories were extracted. Differences between actual and estimated were calculated. The relative effort of each affected class was computed.

Cm_i	US_j	Cl_k	LOC	LOC'	Added LOC	Deleted LOC
Cm_1	US_1	Cl_1	100	120	20	
		Cl_2	150	110		-40
		Cl_5	50	80	30	
Cm_2	US_2	Cl_1	120	95		-25
		Cl_2	110	90		-20
Cm_3	US_1	Cl_2	90	105	15	
		Cl_3	35	35	0	
		Cl_5	80	65		-15
Cm_4	US_3	Cl_1	95	90		-5
		Cl_3	35	55	20	
Cm_5	US_1	Cl_1	90	75		-15
		Cl_5	65	17		-48
Cm_6	US_2	Cl_3	55	70	15	
		Cl_2	105	185	80	
Cm_7	US_3	Cl_2	185	64		-121

Aggregating the commits for the same user story, gives the total change (in terms of LOC) of each affected class in that user story:

	Cl_k	Added LOC	Deleted LOC
$US_1 = Cm_1 + Cm_3 + Cm_5$	Cl_1	5	
	Cl_2		-25
	Cl_3	0	0
	Cl_5		-63
$US_2 = Cm_2 + Cm_6$	Cl_1		-25
	Cl_2	60	
	Cl_3	15	
$US_3 = Cm_4 + Cm_7$	Cl_1		-5
	Cl_2		-121
	Cl_3	20	

Figure 4.4: Output of A.1 and A.2 - Example illustration

Since effort is generally estimated for the user story, a means for finding the relative effort of each affected class in that user story was needed. Therefore, a metric-based algorithm was devised. Example weights were assigned to the code metrics. The implementation steps of the algorithm are as follows:

- A.3.1: Before finding the relative effort, the relative change (RC_i) of each class in a user story was found. The relative

Assumed old user stories in DB with affected classes, impact factors and actual effort as follows:

Old US _n	CIn	LOC	CBO	WMC	Actual effort (AE)	Relative effort (REC)
US ₁₅	Cl1	20	50	30	10	2.853
	Cl2	10	60	40		3.156
	Cl3	30	70	40		3.989
US ₂₁	Cl1				5	
	Cl5					
US ₈	Cl2				4	
	Cl7					
	Cl8					

To calculate the relative effort of each class in a user story:

A.3.1: Calculating the relative change (RC) of each impact factor through: $RC_i = \frac{Y_i}{\sum_{i=1}^{TC} Y_i}$

For example for LOC in US₁₅ it is: $\frac{LOC\ Cl_1}{LOC\ Cl_1 + LOC\ Cl_2 + LOC\ Cl_3}$

Relative change of LOC, CBO and WMC resulting in:

		Relative change		
		LOC	CBO	WMC
US ₁₅	Cl ₁	0.33	0.28	0.27
	Cl ₂	0.17	0.33	0.36
	Cl ₃	0.50	0.39	0.36

A.3.2: Calculating the weighted sum of the relative changes of all affected classes in a user story through:

$$\frac{\sum_{i=1}^{TC} W_i * RC_i}{\sum_{i=1}^{TC} W_i}$$

Assumed weights of the impact factors: LOC =1; WMC = 2; CBO =3

Resulting in:

Weighted sum of relative change		
US ₁₅	Cl1	0.2853
	Cl2	0.3156
	Cl3	0.3989

A.3.3: Calculating the relative effort of each class in the user story through:

$$REC_i = \frac{\sum_{i=1}^{TC} W_i * RC_i}{\sum_{i=1}^{TC} W_i} * AE_j$$

Resulting in:

Relative effort of each class		
US ₁₅	Cl1	2.853
	Cl2	3.156
	Cl3	3.989

Figure 4.5: Output of A.3 - Example illustration

change means how much a certain class has contributed towards the total change for a certain user story regarding each metric (Y_i) across the total of all affected classes (TC), calculated as follows:

$$RC_i = \frac{Y_i}{\sum_{i=1}^{TC} Y_i}$$

- A.3.2: Then the weighted sum of the relative changes of all affected classes in a user story was calculated.

$$\frac{\sum_{i=1}^{TC} (W_i * RC_i)}{\sum_{i=1}^{TC} W_i}$$

where W_i is the weight of the corresponding code metric (Y_i), RC_i is the relative change of the i^{th} class.

- A.3.3: The weighted sum was multiplied by the effort of the associated user story (AE_j). This gave the relative effort of each class (REC_i).

$$REC_i = \frac{\sum_{i=1}^{TC} (W_i * RC_i)}{\sum_{i=1}^{TC} W_i} * AE_j$$

where AE_j is the associated effort of the j^{th} user story, and REC_i is the relative effort of each i^{th} class affected by the j^{th} user story.

This relative effort is a means to support experts in roughly assessing the associated effort a certain potentially impacted class may have in a specific user story. All the user story and class level impact information along with the respective actual and calculated relative effort was stored in the created impact analysis model for performing impact analysis. An example illustration is given in Figure 4.5. It shows an excerpt of the output of A.3. (i.e., calculation of relative effort of each affected class).

Output: IA model with added historical impact analysis information. This information was combined and visualized through the developed prototype to support experts in making estimates (see C.2 and C.3 in the workflow Figure 4.7).

4.6 Estimation model development - Process B

Boosting is a non-linear regression procedure in which weak classification algorithms are sequentially applied to the incrementally changed data, to create a series of decision trees that produce an ensemble of weak prediction models. While boosting trees increases their accuracy, it also decreases speed and human interpretability. On the other hand, gradient boosting procedures generalize tree boosting to minimize these issues [4]. Boosting such as stochastic gradient boosting (SGB) has already been applied for predicting development effort [85] [101] and has outperformed other models based on neural networks and random forests by achieving higher prediction accuracy [100].

GBT is the ensemble of either regression or classification tree models that use forward-learning ensemble methods to obtain predictive results through gradually improved estimations. Furthermore, ensemble methods provide more accurate prediction accuracy compared to individual models and hence are more reliable [47]. This motivated the choice of using GBT for building the estimation model.

The estimation model was developed using the following sub-processes. It is shown as process B in Figure 4.3.

1. B.1 Dataset:

Input: The model utilizes a subset of the same dataset that was used to create the impact analysis model in Sec 4.5.

Process: The dataset comprised a total of 345 user stories.

In the dataset, seven parameters were identified and collected for implementation purposes. The parameters demonstrate the total number of changes (additions and deletions) in LOC, CBO, and WMC of class/es and the number of classes affected when a single user story is implemented. The last parameter is the actual effort spent on finishing the development of the user story. This dataset was used to predict the total effort required to develop a user story in a sprint.

Output: Parameters.

2. B.2 Modeling method:

Input: Dataset classified into parameters.

Process: Depending on the case and data one may have to consider assumptions when choosing an approach. For example, first, we considered multivariate linear regression. Multiple regression assumes that the independent variables are not highly correlated with each, i.e., no multicollinearity. Similarly, it also assumes that the variance of error terms are similar across the values of the independent variables, i.e., Homoscedasticity. But in our case, there was multicollinearity among the independent parameters, and homoscedasticity was not met either. Moreover, the data was highly right-skewed; therefore, the data processing was needed comprehensively to apply multivariate linear regression. As an alternative, we considered GBT regression models, as they are capable of handling conditions such as multicollinearity by themselves. GBT is also an example of ensemble methods that provide more accurate prediction accuracy compared to individual models, and hence are more reliable [47].

Output: Chosen modeling method which in this concrete case turned out to be a GBT model.

3. B.3 Data normalization:

Input: Dataset classified into parameters.

Process: It was observed that the dataset was not normally dis-

tributed (based on the skewness and kurtosis values). Hence, to make the data normally distributed, logarithmic transformation was applied over the dataset.

Output: Normalized dataset.

4. B.4 Model creation:

Input: Normalized dataset.

Process: The model was implemented using RapidMiner¹. First, the dataset was divided into a training set and a test set using a ten-fold cross-validation technique. Since log transformations were applied to the data set, the data type became "real" and thus, regression was performed. Then the variables to control the complexity of the trees were specified. Some of the default variables are as follows: 1) Number of trees = 20, 2) maximal depth = 5, 3) number of bins = 20, 4) learning rate = 0.1 etc. GBT requires careful tuning of the variables as they may overfit the training data. We can generate less complex trees by setting the number of trees variable to a lower value, e.g., 5 instead of 20 and also slower the learning rate, e.g., 0.01 instead of 0.1. This will create less complex shallow trees and less overfit data. However, this requires some trials before finding an optimal model. Nonetheless, we created the model using default values. The GBT model was then created and applied to the test data. The performance of the model was calculated by using evaluation metrics described in Section 5.4.4. , Afterward, the model was recreated on the whole dataset to be applied on the estimation data (test data in this regard) gathered during the evaluation workshop.

Output: GBT model. The output of one of the 20 trees (Tree 1) generated using default values is shown in Figure 4.6. The nodes represent the corresponding model parameter and edge their respective values. In this Figure 4.6, Tree 1 begins with the AddedWMC parameter, and if the added complexity (AddedWMC) is greater than or equal to 1.020, it goes to the right side of the tree and checks the value of DeletedLOC parameter and so on.

4.7 Detailed work flow - Process C (Perform estimation)

The work flow of the method encompasses the "Process C" as shown in Figure 4.7. The workflow begins when a new user story is to be estimated by the experts in an estimation meeting.

The details are explained as follows:

- C.1 Impact set identification:
Input: User story, experts.

¹<https://rapidminer.com/>

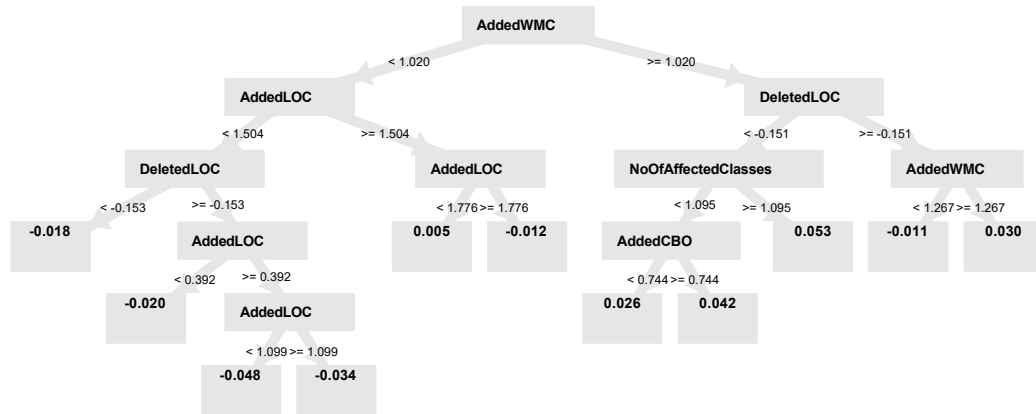


Figure 4.6: GBT model - Tree 1 view

Process: In this process, the experts, identify the initial “impact set” (the “seed” for triggering change IA) based on their opinion after analyzing the user story description and doing discussion. This impact set is composed of class/es that the experts identify as candidates that will potentially be impacted due to the implementation of this user story.

Output: Identified potentially impacted code parts (class/es).

- C.2 Apply impact analysis:**
Input: Impact set (seed), Impact model (generated from process A. Impact analysis model development process).
Process: Through the selected static and historical impact analysis techniques, impact analysis is performed. Static impact analysis returns the quantified impact through code metrics, and historical impact analysis returns the effort data, other affected classes, co-changed classes as mentioned in Section 4.3. The analysis results are visualized in the form of tables, charts and dependency graphs providing structural dependence information.
Output: Analysis results.
- C.3 Revise impact analysis results:**
Input: Analysis results, experts.
Process: The experts can interact with the visualized impact analysis results and make selections regarding the potential impact of the selected user story.
Output: Revised analysis results.
- C.4 Estimation:**
 This is a decision point in the workflow. It has two paths, where the experts choose the path to perform the estimation. One path leads to hybrid estimation and the second leads to estimation with the GBT model.

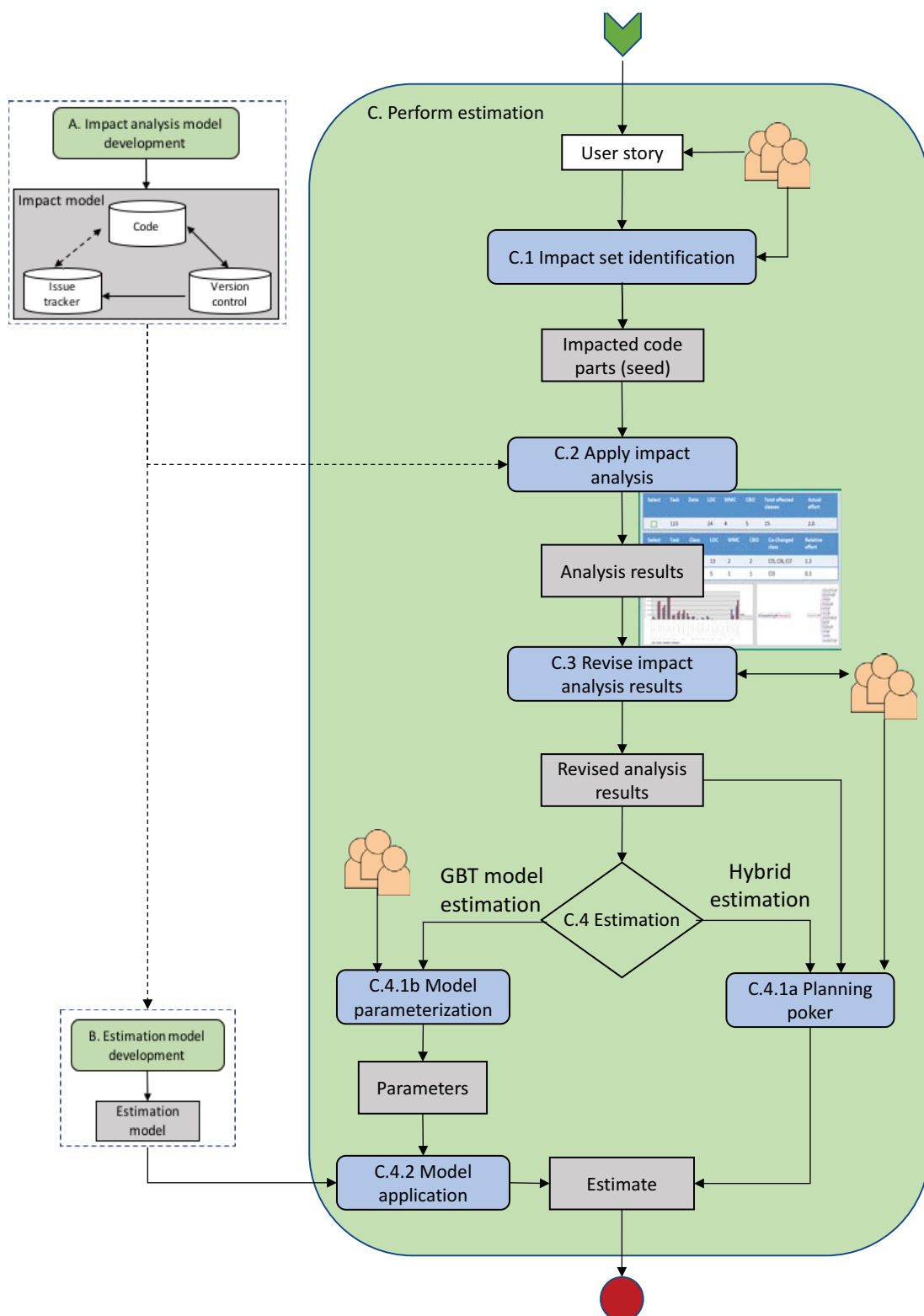


Figure 4.7: Detailed work flow of Process C

- Hybrid estimation: If experts choose this path, they would consult, discuss the revised impact analysis results provided to them in an expert based estimation method like Planning Poker and then provide an estimate for the user story.

- GBT model estimation: If experts choose this path, they would like to use GBT model estimates by supplying the model parameters. The model provides them with an estimate for the user story.
- C.4.1a Planning poker:
Input: Revised analysis results, experts, user story
Process: The experts will take into account the revised analysis results and play Planning Poker to estimate effort for the user story.
Output: An estimated effort of the user story.
- C.4.1b Model parameterization:
Input: Experts.
Process: The experts will provide input to GBT model parameters.
Output: Parameters of the model are provided.
- C.4.2 Model application:
Input: Estimation model (generated from process B. Estimation model development process), model parameters.
Process: The model is then applied to obtain an estimate for the user story.
Output: An estimated effort of the user story.

The workflow ends once the user story is estimated. The whole information comprising the impact model, expert estimation and the estimation model is stored in an experience base, which can be updated with current sprint data to support estimation for the next sprint data. With this experience base, organizational learning is enabled at the project level.

To summarize, the impact information is visualized for the experts such that they can interact with it to find the revised potential impact along with the historical effort data. With the help of this information, the expert is asked to perform estimations for the given new user story. The experts can also use the GBT model to obtain an estimate for a given user story. The estimate is saved and also serves as test data for applying the estimation model in the next sprint.

4.8 Estimating a user story using HyEEASe - an example

In this section, a hypothetical example is described to demonstrate the process of estimating a user story using HyEEASe. This example considers a system, i.e., Moodle [13]: modular object-oriented dynamic learning environment. It is an open source, online learning management system. Details regarding the Moodle platform and its architecture can be found in Appendix F.

In this hypothetical example, a user story US_n is assumed that requires enhancement of existing functionality, i.e., "Plagiarism prevention". In Moodle, plagiarism is detected through "plagiarism prevention plugin"

which a) currently supports the assignment activity and b) is only able to detect whether the text in an assignment is copied from a source on the Internet. The user story US_n requires an enhancement, that in addition to detecting the copied text, the system also generates:

- Report (with statistics) indicating what parts of the assignment have been copied.
- Analyze the plagiarized part and suggest the action category (acceptable, acceptable after revision, not acceptable).
- Analyze the plagiarized part and visualize affected parts with percentages (graphs).

Now let us further assume, some experts are estimating US_n using the HyEEASe prototype. A simplified version of the HyEEASe prototype UI is shown in Figure 4.8. The process is explained as follows:

- C.1 Impact set identification: The experts have read US_n description and needed to identify the impact set. In this case, the experts identified the seed as "plagiarism plugin". The seed was given as input to HyEEASe in the "Enter seed" field in Figure 4.8.

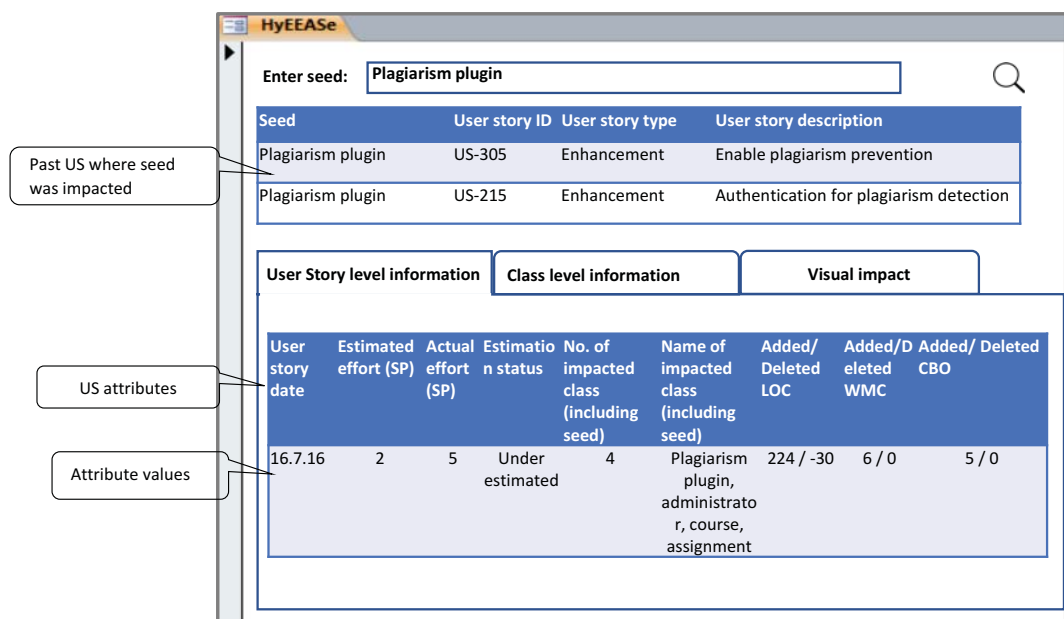


Figure 4.8: Output of C.1 Impact set identification and C.2 Apply impact analysis

- C.2 Apply impact analysis: After the identification of the impact set, while using the HyEEASe prototype, the expert applied impact analysis. This step utilized the impact model developed in process A.

HyEEASe performed impact analysis on Moodle code using the “plagiarism plugin” as input. As a result, the returned impact is displayed by HyEEASe to the experts. This impact comprised of information based on the given “seed”, i.e., the “plagiarism plugin”. The impact information contains all the corresponding previously implemented user stories where the “plagiarism plugin” was impacted (see “Past US where the seed was impacted” in Figure 4.8). In this case, when “plagiarism plugin” was given as seed to HyEEASe, it found two previously implemented user stories, i.e. “US-305” and “US-215”. US-305 was about enabling plagiarism prevention, and US-215 was about authentication for plagiarism plugin.

- C.3 Revise impact analysis results: The experts interacted with the visualized impact analysis results and made selections regarding the potential impact of the selected user story. Upon selecting the user story US-305, HyEEASe displayed the corresponding user story level impact information that comprised of “US attributes” for US-305 with values as seen in “Attribute values” (see “User story level information” tab in Figure 4.8).

In this tab, HyEEASe informed the experts that US-305 was estimated with 2 SP but took 5 SP and so was underestimated. This implementation impacted four components in Moodle code, i.e. “Plagiarism plugin” (seed), administrator, course, assignment”. Due to this implementation, impact factors were also significantly changed across all of the above mentioned four impacted components in Moodle code i.e., altogether 224 lines of code (LOC) were added, 30 were deleted, cyclomatic complexity (WMC) was also increased by 6 units and coupling between objects (CBO) was increased by 5 units.

Furthermore, HyEEASe also displayed class level impact information that comprised of “Class attributes” and their values in “Attribute values” for “plagiarism plugin” (see Figure 4.9, “Class level information” tab).

In this tab, HyEEASe informed the experts that due to US-305, when “plagiarism plugin” was changed, it individually took 2 SP effort out of the 5 SP total effort of the US-305. This implementation also changed the impact factors significantly in the “plagiarism plugin” in the Moodle code, i.e., Altogether 50 lines of code (LOC) were added, 15 were deleted. Cyclomatic complexity (WMC) was increased by 4 units, and coupling between objects (CBO) was increased by 5.

HyEEASe also visualized the impact information through a variety of charts (bar/ column/ pivot charts), (see “Visual impact” tab “User story level impact factors” (left) in Figure 4.10). With this information in one holistic view, the experts analyzed all the changed impact factors for all previously implemented user stories (i.e., US-

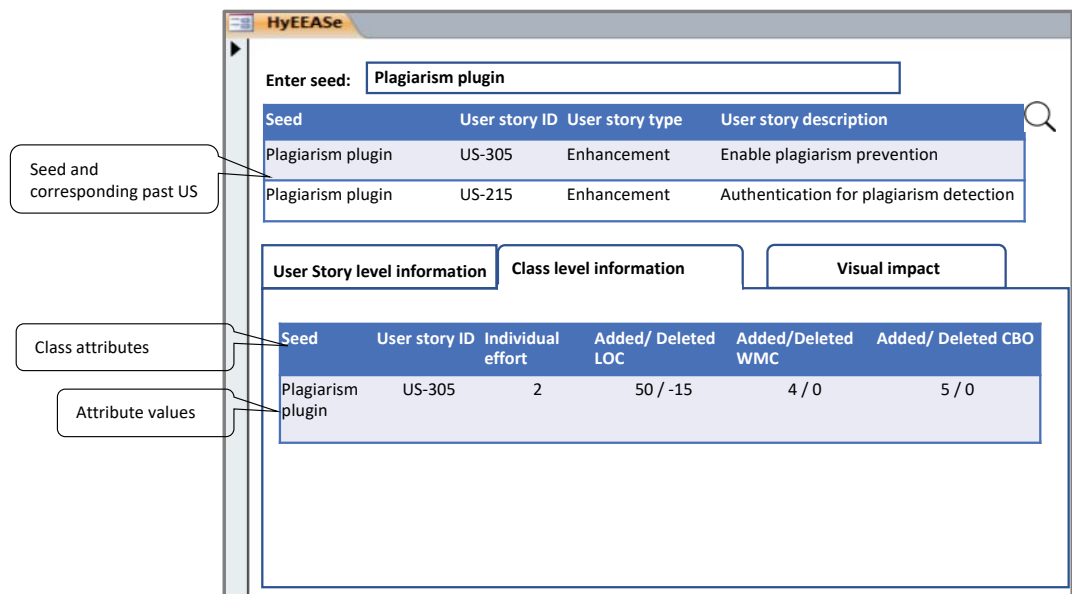


Figure 4.9: C.3 Revise impact analysis results - 1

305 and US-215) where “plagiarism plugin” was one of the impacted components among others. In the case of US-305, the chart shows that actual effort was 5 SP, added LOC was 224, deleted LOC was 30, added WMC was 6 and added CBO was 5 across all 4 impacted components (Plagiarism plugin, administrator, course, assignment).

Additionally, HyEEASe displayed dependency graphs, showing the code internal structure, of impacted seed component (see in Figure 4.10, “Visual impact” tab “Dependency graph” (right). While exploring, the experts found out that, components like “authentication, teacher, student, assignment and grade” were tightly coupled with “plagiarism plugin”.

The experts while discussing can select/deselect which of the previously impacted classes may potentially be impacted again while implementing the US_n . Based on the expert’s selection, HyEEASe re-performed the impact analysis and updated the charts and dependency graphs, calculated/recalculated the individual effort of the associated classes and displayed to the experts.

- C.4 Estimation: The experts first chose Hybrid estimation path followed by GBT model estimation.
 - Hybrid estimation: Experts had all the information which they consulted, discussed and revised impact analysis results provided to them. They provided an estimate for the new user story US_n , e.g., 8 story points.

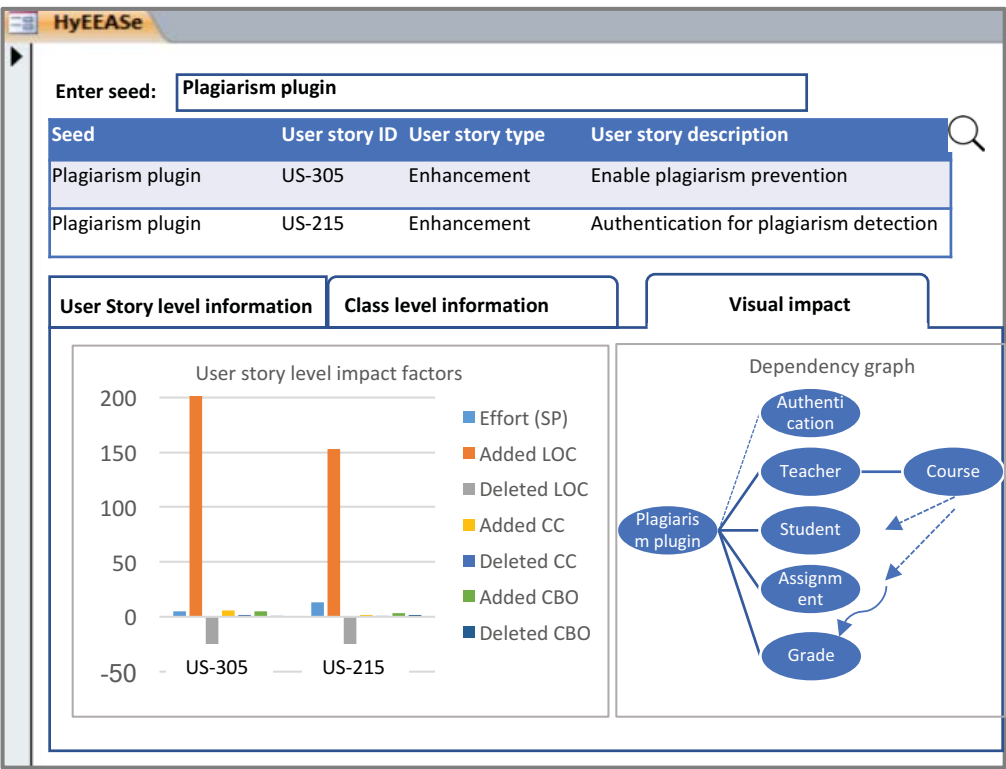


Figure 4.10: C.3 Revise impact analysis results - 2

- GBT model estimation: After Hybrid estimation, the experts used GBT model estimation. This step utilized the estimation model developed in process B. The experts supplied the model parameters using the impact information. Let us assume they provided the following input parameters as shown in Table 4.2. The model provided them with an estimate, e.g., 6 story points for US_n .

Table 4.2: GBT estimation model parameters input - an example

No. of affected classes	Added LOC	Deleted LOC	Added WCM	Deleted WCM	Added CBO	Deleted CBO
3	26	-18	2	0	7	-4

It is up to the experts to reconsider their estimates after seeing the GBT model estimation or leave them as is till the end of the sprint and compare with the actuals for learning purposes. This example illustrated that with HyEEASe, the expert is provided with all the information of identified impact set along with the relative efforts. Additionally, the visualizations in the form of charts and dependency graphs supported the experts during estimation.

4.9 Assumptions /applicability of HyEEASe

There are certain limitations of HyEEASe. The change impact analysis would effectively work if:

1. The system or software to which the change requests (user stories or backlog items) are being made is partially developed means the base code exists, and functionality is being added to it incrementally in each sprint.
2. The change requests are not completely independent of the existing system.
3. Historical information regarding the actual/ estimated effort data exists.
4. Both Hybrid and GBT model estimation currently only consider a limited number of code metrics like impact factors or model parameters respectively, i.e., Size in LOC/SLOC, Cyclomatic Complexity (WCM), and Coupling Between Objects (CBO). This is not an exhaustive list of impact factors and therefore requires further research to find more appropriate impact factors that influence estimation.
5. The developed estimation model is based on GBT which is not robust against incomplete input, therefore, in the ideal case, the input parameters of the models should not be left empty for the model to work effectively.
6. In case of a different organizational context, new change IA techniques might need to be configured for the Hybrid estimation to work. In that case, the GBT model also needs to be re-calibrated on a new data set. The experience base needs to be maintained for the historical data, estimation model and impact model.
7. Hypothesis H5: HyEEASe enables learning of associated effort (from sprint to sprint). HyEEASe enables feedback across multiple iterations to learn about discrepancies between estimated and actual effort. This hypothesis could not be tested as it required a longitudinal study. This, however, is considered for future work.

4.10 Summary

This chapter reported the workflow, design, and development of HyEEASe to support experts while estimating the effort in an agile development context. The method initially was developed in close collaboration with SAP and later was refined by working with Insiders Technologies. Data of seventy-two sprints from Insiders Technologies was gathered, and change impact analysis was applied to it. The method used static and historical IA as well as past estimation data for estimating effort for

a change. GBT model was developed using a subset of the data gathered from Insiders Technologies and was used to predict the total effort for a new user story with similar potentially impacted classes.

5 Empirical evaluation

5.1 Introduction

In this chapter, several evaluations of the proposed method using different empirical methods are reported. First, the findings of the case study [108] with SAP SE where the perceived usefulness of an instance of HyEEASe through mock-ups was evaluated.

Second, the findings of the case study [109] with Insiders Technologies GmbH are reported. Here the effectiveness, the perceived usefulness and the learnability of the refined HyEEASe were evaluated.

Third, the findings of the data-based evaluation are reported where the GBT model is compared with Agile COCOMO II [33] in terms of estimation accuracy.

Fourth, the findings of a controlled experiment are reported. The experiment was conducted with the BS and MS students taking Software Process and Project Management (SPPM) course at the Department of Computer Science, Technische Universität Kaiserslautern (TUKL). The experiment was conducted to find the usefulness and learnability of HyEEASe in comparison to Planning Poker.

The research done for this chapter was conducted as part of research projects, Abakus, grant number 01IS15050G, and HyEEASe, grant number 01IS12053 funded by the German Ministry of Education and Research (BMBF). The results were also published [107], [108], and [109].

This chapter is structured as follows: Section 5.2 describes the evaluation strategies that were used to evaluate the research hypotheses. Section 5.3 describes the case study conducted with SAP SE. Section 5.4 highlights the case study conducted with Insiders Technologies GmbH. Section 5.5 represents the comparison of the GBT model and Agile COCOMO II. Section 5.6 provides details of the controlled experiment. Finally, Section 5.7 summarizes the chapter.

5.2 Evaluation strategies

The main objective of an empirical evaluation is to evaluate the proposed solution whether it solves the problem it is intended to solve. It means that the proposed solution provides certain benefits at some yet acceptable cost when applied in a real context. Empirical evaluation can be done by applying one or a combination of different evaluation strate-

gies including case study, survey, controlled or quasi-experiment [118]. In this thesis, different evaluation strategies are used for evaluating the HyEEASe method, i.e., two industrial case studies, a controlled experiment and a data-based evaluation where the GBT model was compared with Agile COCOMO II estimation model. Figure 5.1 shows a mapping

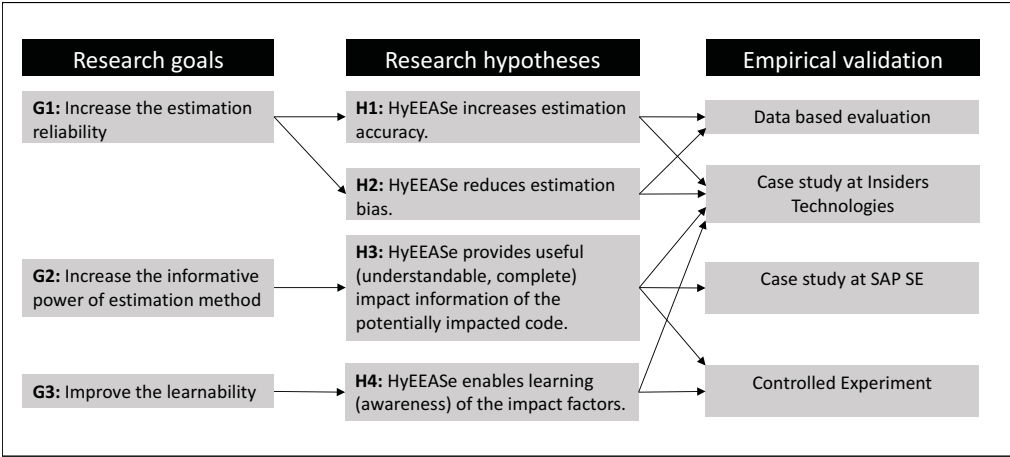


Figure 5.1: Empirical evaluation strategies

of the research goals, hypotheses and the evaluation strategies. The following sections will describe each of the conducted empirical studies for evaluating HyEEASe.

5.3 Case study at SAP SE

This section describes the case study conducted with SAP SE. HyEEASe was developed incrementally, and in this first increment, HyEEASe was comprised of the Hybrid estimation path only and was demonstrated using mock-ups. In this case study, the perceived usefulness of HyEEASe was evaluated using an example scenario. The scenario demonstrated the application and integration of a change IA technique with Planning Poker. The example scenario was presented to the practitioners and was evaluated through mock-ups.

The subsequent sections describe the example scenario, context and selected IA and EE techniques, evaluation goals, hypothesis, design, execution and the results with discussion.

5.3.1 Example scenario - context and selected techniques

In this example scenario, it was assumed that certain experts were estimating efforts for backlog items in a Scrum session using “Planning Poker” as the estimation technique for project planning. Planning Poker

was selected as an estimation technique as it is found to be the most frequently employed expert-based method [114].

Following additional assumptions were made regarding the context:

- A textual change request with requirements (user stories or backlog items) and the source code is available.
- The source code contains comments and identifiers made by the developers. This means textual impact analysis can be made.
- An execution trace of a change request can be generated. This means dynamic impact analysis is also possible.

The conceptual framework (Section 4.2) was utilized to select a change IA technique based on the given example context. With this given context, the change IA technique proposed by Gethers et. al [50] was selected.

The proposed IA technique combines information retrieval (IR), with dynamic analysis and mining software repositories (MSR) to determine the impact set of a given change request. This technique can be applied in three modes for obtaining an impact set [50]:

- Highest degree of automation and the lowest degree of developer supplied information: In this mode only a textual change request is available. In such a scenario, the impact set can be achieved by taking the textual view of source code and applying IR techniques, (e.g., Latent Semantic Indexing or simply, LSI).
- Medium degree of both automation and developer supplied information: In this mode, in addition to the textual change request, a relevant source code entry "seed" entity verified by an expert is also available. A developer somehow narrows down to at least one verified entity that needs a change (e.g., from previous experience of performing similar changes). In such a scenario, the impact set can be obtained by mining (i.e., MSR) the past commits (change history) of software entities.
- Lowest degree of automation and the highest degree of developer supplied information: In this mode, in addition to the textual change request, an execution trace of the change request is also available. A developer has executed the feature, inferred by reading the textual change request, and collected the runtime information - executed methods, (e.g., to verify if the issue that was reported can be replicated or collected from the call stack of a failure). In such a scenario, the impact set can be obtained by applying dynamic analysis and getting the methods executed in the run-time scenario.

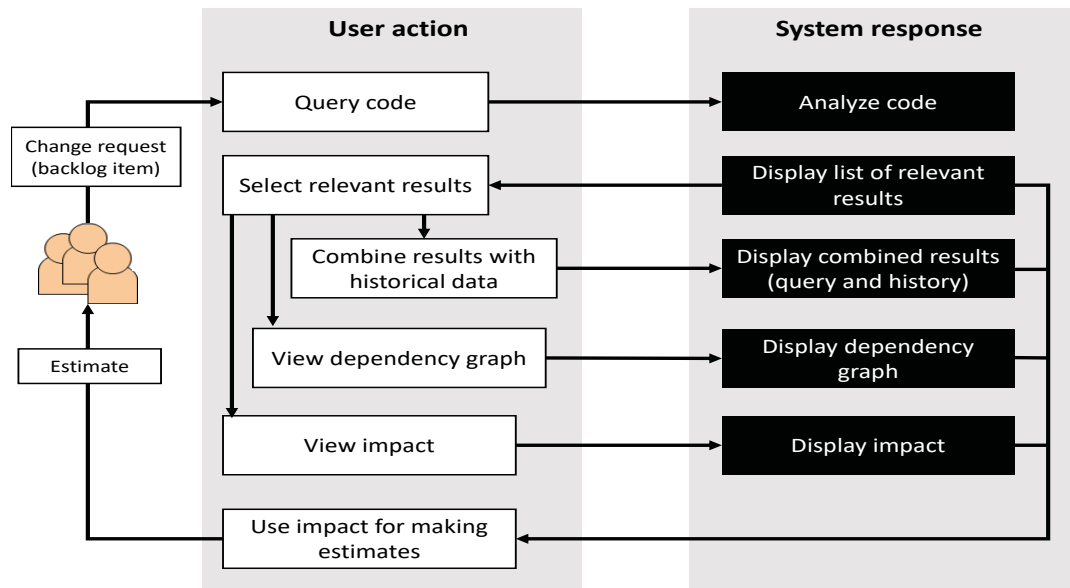


Figure 5.2: Mock-up workflow

When a seed entity is available along with the change request, a combination of IR and MSR is engaged. Similarly, when the dynamic information is available along with the change request, a combination of IR and dynamic analysis is selected. The premise of this technique is that any combination that involves the additional developer supplied information and (highest or medium) automation would provide a better impact set than those based on automation alone. Further details can be found in [50].

5.3.2 HyEEASe mock-up

The mock-up was designed using the Balsamiq tool [2] and exhibited the first two modes of the selected technique as an example scenario. In this section, the workflow of the scenario (Figure 5.2) and the mock-up user interface (Figure 5.3), are described simultaneously:

1. **User action - Query code:** In this step, the user (expert) picks a change request (backlog item) and forms a query from the description. After clicking the “search” button, the query is run on the code, i.e., triggering the first mode of the technique. This step is marked as 1 in Figure 5.3.
2. **System response - Analyze code:** In this step, the system (internally) processes and analyzes the code, based on the query to find the impact set (methods) relevant to the query. This step is marked as 2 in Figure 5.3.

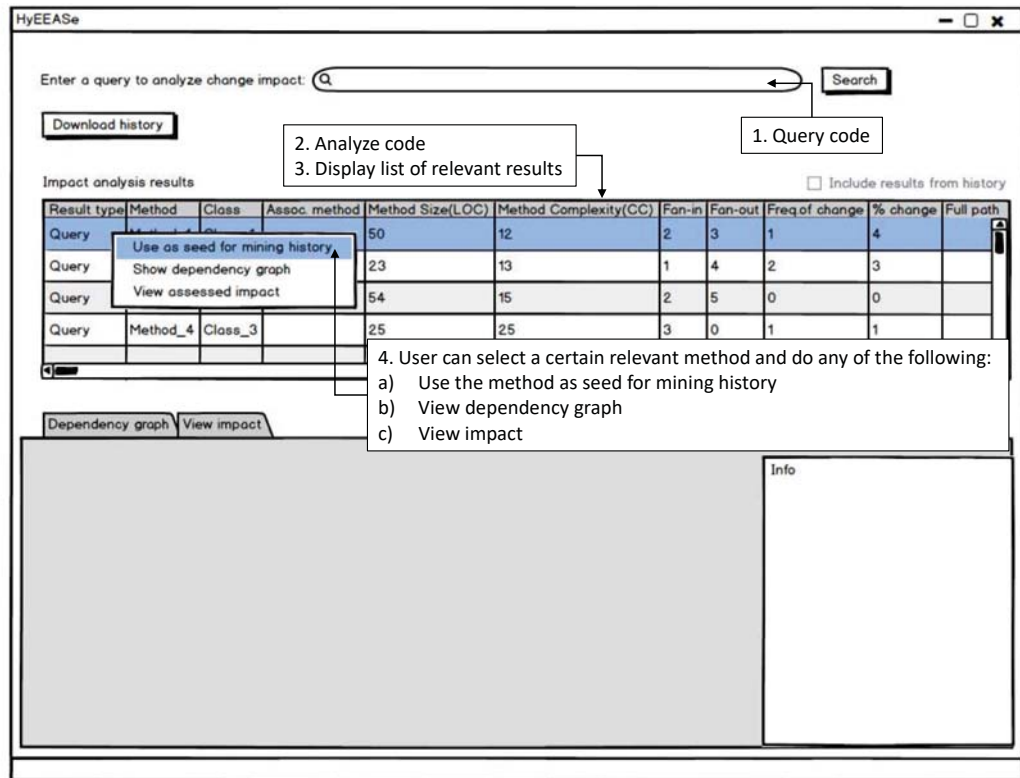


Figure 5.3: Mock-up user interface - initial screen

3. System response - Display relevant results: In this step, the system then displays the impact set, i.e., a list of relevant methods to the user in a table.

This step is marked as 3 in Figure 5.3.

4. User action - Select relevant results: This step is marked as 4 in Figure 5.3. The user can select a certain relevant method and do any of the following:

- User action - Combine results with historical data: The user can use the selected method as a "seed", to combine query results with historical data (perform MSR), i.e., triggering the second mode of the technique.
- System response - Display combined results (query and history): The system displays the combined results (impact set) of query and history in the same table.

This step is marked as 4a in Figure 5.4.

- User action - View dependency graph: The user can view a dependency graph of the selected method.

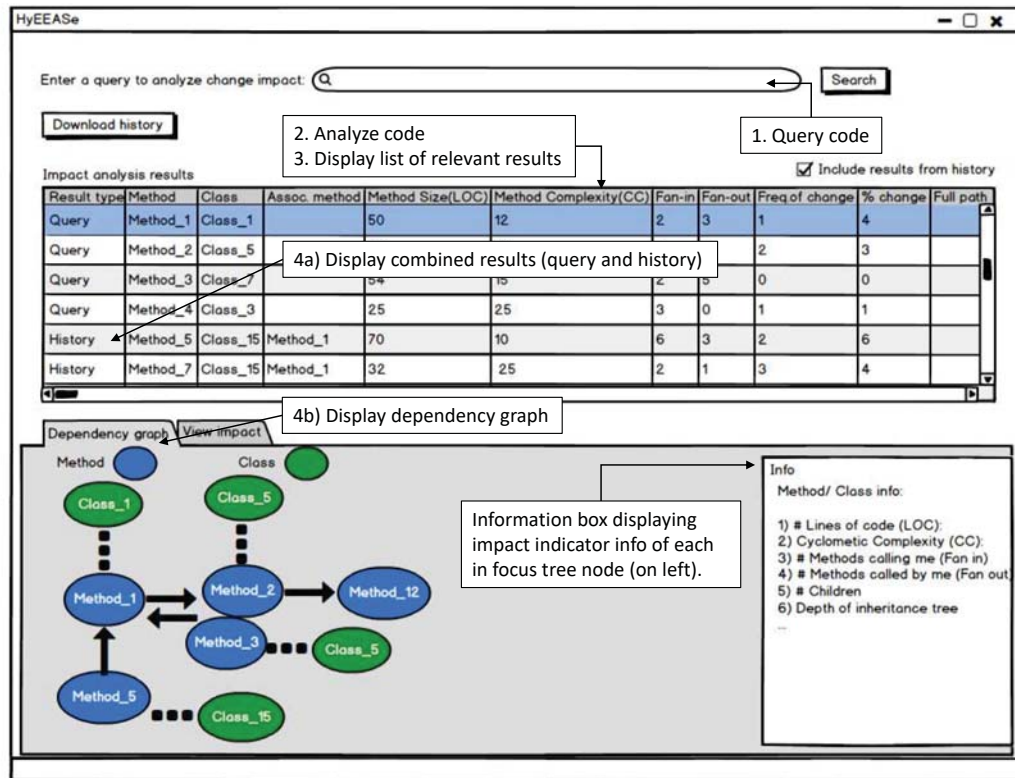


Figure 5.4: Mock-up user interface - dependency graph view

- System response - Display dependency graph: The system displays a dependency graph showing information on every (in-focus) tree node in an information box.

The resulting dependency graph can be seen as 4b in Figure 5.4.

- User action - View impact: The user can view the impact of the selected method/s.
- System response - Display impact: The system shows the impact of the selected method/s for the selected impact metrics.

The resulting impact can be seen as 4c in Figure 5.5.

It is important to mention that as a result of the case study conducted with SAP during the state-of-the-practice analysis (see Section 2.3) the experts realized the importance of the impact information and considered it useful for effort estimation. That impact information comprised of certain factors (see Table 2.2), out of these, few were code-based impact related factors. These factors were quantified into code metrics as seen in Table 4.1.

Few other code related impact factors were also presented to the experts as examples i.e. 1) The frequency of change.

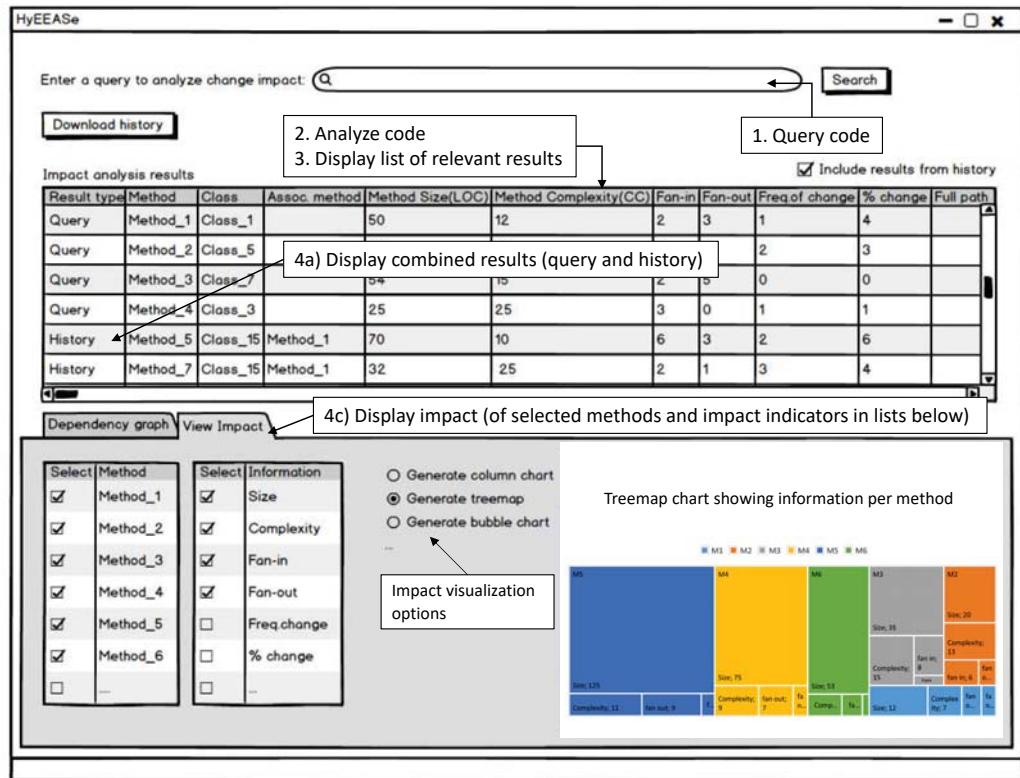


Figure 5.5: Mock-up user interface - impact view

It refers to the frequency of change of the potentially impacted methods/classes in the version history. 2) Percentage of change. It refers to the percentage of change of the potentially impacted methods/classes in the version history. The purpose was to observe in the feedback session if these code-based impact factors and metrics would be sufficient for the experts or would they require other code-based impact factors and/or metrics to quantify them.

5. User action - Use impact for making an estimate: Aggregating all three options (combined history data, dependency graph, impact information), the user can utilize this information when estimating the effort for a certain backlog item.

5.3.3 Evaluation approach

In the following sub-sections, the evaluation goals, design, target population, execution procedure, data collection, and analysis are described.

The case study was conducted to provide a piece of empirical evidence that HyEEASe achieves its research goals when evaluated in a real industrial setting. In particular, the research goal G2 and corresponding hypothesis H3 were evaluated as seen in Figure 5.1. In particular, we

wanted to prove that the concept and the supported mock-up provide useful impact information.

Using the GQM template, the goal is formulated as: "To understand and analyze HyEEASe and mock-up with respect to its perceived usefulness from the perspective of agile development teams in the context of effort estimation in agile software development."

5.3.4 Evaluation criteria

To evaluate H3 (usefulness) of HyEEASe and mock-up, quality aspects like "understandability" and "usefulness" were evaluated from an instrument provided by McKinney et al. [78]. Certain other related aspects like "perceived ease of use", and "behavioral intention" were evaluated using TAM [116], and "applicability" and "appropriateness" from Lee et al. [68]. The quality aspects considered are shown in Table 5.1.

5.3.5 Evaluation design - sample and population

As mentioned earlier, the evaluation was conducted in a case study setting jointly with SAP SE. The same three agile development teams at SAP were contacted (Section 2.3.2) with whom the case study [106] was done because they were the right stakeholders and also the end users of HyEEASe; therefore their feedback is the most important one. However, altogether only six participants from all three teams with various roles were able to participate. Two of these six participants were product owners, one was both the scrum master and a developer, two were architects, and one was a developer. The aggregated experience of all the participants in software development was 16.5 years (median value) and in agile development 7.75 years. Regarding the context, domain, type of product, etc. please refer to Section 2.3.2.

5.3.6 Execution

The protocol used in this evaluation includes:

- The procedures for inviting participants and making arrangements for the fieldwork.
- The informed consent form that allowed to collect data for the intended analysis and informing the participants about the evaluation as well as their rights.
- The data collection procedures and data analysis procedures.
- An interactive workshop was conducted where the example scenario and the mock-up could be presented, and simultaneously the evaluation could be performed.

- Another interactive feedback session was designed and executed at the end of the workshop. The materials can be found in Appendix C.

This evaluation lasted two hours. During the whole evaluation, two researchers were present; one was the moderator, and the other was an observer and took notes.

5.3.7 Data collection and analysis

The quantitative data was collected using a questionnaire designed by using TAM[116] and the instrument provided by McKinney et al. [78]. Using the questionnaire, the participants rated the quality aspects of HyEEASe and the mock-up. The qualitative data was collected using moderation cards where the participants provided their open feedback (see Appendix C). The quantitative and the qualitative data were analyzed, compared, and integrated the results for both evaluation sessions. For the quantitative analysis, descriptive statistics including the median value are reported. For the qualitative data, themes were derived from the feedback session.

5.3.8 Results and interpretation

This section presents the analysis and results of the evaluation and discusses them simultaneously.

The results of the evaluated hypothesis (H3) are shown in Table 5.1. The practitioners were asked their level of agreement on the quality aspects (scale: five-point ratio scale, with 5 being strongly agree and 1 being strongly disagree). Quality aspects rated with more than 3 (neutral opinion) show a positive evaluation overall.

Overall, the participants found HyEEASe understandable and useful (median value = 4 in Table 5.1). However, regarding its applicability, the participants were neutral (median value = 3 in Table 5.1). This might be because during the feedback session the participants reflected that they found the query formulation difficult as it depends on the developer's familiarity with the underlying system (see improvement suggestions in Table 5.3).

Moreover, in their view, HyEEASe did not take into account certain aspects such as New development or Test coverage which makes them unsure regarding its applicability. It was planned to incorporate the Test coverage in the next iteration. However, regarding New development, it must be emphasized that code-based IA techniques are built on a premise, to find the potential impact, the system has been developed, and the change requests are to some extent related to the existing code and are not completely independent of it.

The participants found the mock-up useful (median value = 4 in Table 5.1) and they also intend to use it if it is implemented as a tool. However, they disagree that the mock-up is appropriate for their work (median value = 2.5 in Table 5.1).

The analysis reflects two possible reasons for this. First, it could be that the practitioners got slightly distracted from the focus of evaluation when they raised their concerns regarding the query formulation in the presented example scenario. The selected IA technique [50] requires a query from the developer upon which it performs the IA, based on the premise that the developer's knowledge is the supreme source of information for the impact to start with. It is to be noted that the focus of the evaluation was not to evaluate any underlying IA technique concerning the inputs/outputs etc. but to evaluate the utilization of IA for EE instead. However, this was important feedback that motivated us to find other comparatively simple IA techniques for the next development increment.

Second, the missing features and constraints indicated by the practitioners in the feedback session, e.g., Test coverage, Dynamic analysis, New development etc., made them think that it is still not very appropriate for their tasks.

The practitioners were asked to provide their feedback regarding Hy-EEASe and the mock-up. The summary of positive feedback with categories is shown in Table 5.2. In general, the participants appreciated the simple and intuitive user interface of the mock-up. They found Hy-EEASe useful to support estimation where no objective information is available, as it provides more objectivity to estimation by visualizing the impact information compared to the subjective nature of expert judgment. Impact information visualization would help them learn which other parts of the system they need to take care of while estimating.

The summary of improvement suggestions, their categories, and explanations by the practitioners along with the response from the author addressing the corresponding suggestion are shown in Table 5.3. Among the improvement suggestions, the participants had their concerns regarding the selected IA technique in the example scenario. For example, they raised the importance of precise query formulation, and its relation to the developer's knowledge as it is the quality of the query that derives the impact set. Their concerns are reasonable and also explained in Table 5.3) that in any IA technique, there will be certain premises, constraints, and limitations that have to be taken into account before making a selection.

5.3.9 Threats to validity

In the following, some threats to the validity and the mitigation strategies are discussed.

Table 5.1: Results of evaluated quality aspects of HyEEASe method and the mock-up (median values)

Quality Aspect	Reference	HyEEASe method	Mock-up
Understandability	McKinney et al.[78]	4	NA
Usefulness	McKinney et al.[78]	4	4
Perceived ease of use	TAM[116]	NA	4
Behavioral intention	TAM[116]	NA	3.5
Applicability	Lee et al.[68]	3	NA
Appropriateness	Lee et al.[68]	NA	2.5

Table 5.2: Positive feedback given by practitioners

Category	Feedback
Usability	Simple user interface. Not too time consuming to use. Easy to understand.
Visualization	Visualization of complexity. Helpful search and visualization options. Visibility of impact. Visual display of code metrics
Estimation support	Supports more objective estimation. Provides assistance for implementing a change while estimation.
Learning	Makes impact understandable to new colleagues. Helps to learn about the whole system. Helpful for new colleagues in case of staff turnover.

- **Construct validity:** To avoid inappropriate measurement of the evaluation quality aspects, existing reliable test instruments as TAM [116] were considered, adapted and used to focus on HyEEASe and the mock-up individually.

To avoid mono-method bias, quantitative as well as qualitative analysis were combined and compared. By involving more than one researcher in each step, the reliability of this study is increased.

- **Internal validity:** To ensure the internal validity, only those teams were approached that had sufficient experience in agile methodologies. To avoid evaluation apprehension, the practitioners were assured of the anonymity of their personal and organizational data by the informed consent provided. Another threat to validity in a workshop setting is that of group dynamics. Group dynamics may bias or hinder the participants from freely expressing their true opinion. In this study, through moderation and with the use of a written questionnaire we attempted to reduce this threat.
- **External validity:** As the evaluation used convenience and small sample, the results might not be generalizable. However, to increase the representativeness of the results, they may be repeat-

Table 5.3: Improvement suggestions given by practitioners and corresponding response from authors

Suggestions by practitioners	Response by authors
Challenges in query formulation	
1) The query must be as precise as possible to get good results. 2) More guidance on query formulation would be helpful.	In the mentioned scenario, the selected IA technique [50] is based on the premise that the developer's knowledge should be sound enough to generate a meaningful query out of the change request (backlog item) description. Nonetheless, other IA techniques directly ask for the identification of a code entity instead, which requires more knowledge of the code and the system than a situation where the developer can capture keywords, maybe from the change request description, to make a meaningful query. Regarding guidance on query formulation, dedicated research has been done on how to formulate a search query, where practitioners may be able to get help. In this scenario and through this mockup, only the results of the query could be shown. Nonetheless, this is a good suggestion to also incorporate concepts from query formulation research into HyEEASe.
Missing features	
Dynamic analysis: HyEEASe does not consider dynamic analysis. Test coverage: HyEEASe does not covers required test cases in the impact.	The selected IA technique [50] is indeed capable of doing dynamic analysis; however, to use IA for supporting EE, the scope had to be reduced to static analysis and combined it with history mining in the initial steps. At this initial stage of finding out whether the quantification of impact in metrics like size, complexity, coupling, etc. supports EE, test coverage was not considered. However, it is planned in the future to do so for the next step towards supporting EE. Besides, this also depends on the capabilities of the selected IA technique and on what is found in the historical data.
Constraints and limitations	
Maintenance effort: Effort required for tool maintenance and pre-requisite like training etc. Non-code based backlog items: Items (artifacts) like requirements, architecture, etc. are not addressed by HyEEASe New development: New functionality that is independent of existing code is not covered by HyEEASe.	It is agreed that if such a tool is implemented, someone has to maintain it and some training is also required. Since HyEEASe is dealing with code-based change impact analysis, so the selected IA is currently restricted to the code artifact, only code-based techniques are considered. Other techniques, e.g., at the requirements or architecture level, are not very mature yet, lacking empirical evidence. Moreover, requirements level IA require formalism to generate the impact, which is effort intensive and does not suit the agile development context either. One of the basic premises of HyEEASe is that to find the potential impact, the system should exist, that the system has been developed and the change requests are to some extent dependent/ related to the existing code and are not completely independent from it.

able in a similar context and with similar characteristics of the participants for agile development.

- **Conclusion validity:** The results of the participants using only median values and the number of practitioners to identify common practices/views and to point out potential future research areas are aggregated. However, due to the small sample, more evaluation studies are needed to increase statistical power. To improve the confidence in the interpretation of the data and validity of the conclusions, we took several steps e.g. a second researcher reviewed the analysis and the results and our interpretations were also presented to the companies for validation.

5.3.10 Conclusion

We have reported the results of a case study evaluating HyEEASe with the associated mock-up concerning several quality aspects. It is concluded that the practitioners perceived HyEEASe overall useful for supporting expert-based estimation. They also found the mock-up easy to use and intended to use it in their estimation process.

Among impact metrics size, complexity, and coupling were considered useful whereas the other factors like frequency/ percentage of change were not considered useful for effort estimation. Furthermore, in their view, considering test coverage metrics along with other impact metrics can further improve the effectiveness of effort estimation.

The practitioners acknowledged that the impact visualization provides more objectivity to estimation compared to the subjective nature of expert judgment. They further agreed that the mock-up provides learning opportunities and would help new employees in comprehending the existing system. They further elaborated that no point estimation methods are required. The practitioners raised their concern regarding using point estimation methods to support Planning Poker. In their view, provision of point estimates could refrain them from discussing pro-actively in the team.

5.4 Case study at Insiders Technologies

This section describes the case study conducted with Insiders Technologies GmbH where the effectiveness, the perceived usefulness, and learnability of the refined HyEEASe were evaluated. The subsequent sections describe the refinements made to HyEEASe (based on the feedback of the initial evaluation with SAP), evaluation objects, goals, hypothesis, design, execution and the results with discussion.

5.4.1 Refinements of HyEEASe

Based on the improvement potential and feedback provided by the practitioners at SAP SE in Section 5.3, the hybrid method was refined, and a prototype based on the case company's context and data was developed. The refinements included a different selection and adaptation of the IA technique for integration in the Planning poker estimation method. It is because the experts at SAP SE raised their concerns that the selected IA technique [50] for the example scenario, was both effort and knowledge-intensive and therefore might not be very useful. This feedback and the changed context (see organizational context 5.4.6) of Insiders Technologies, motivated and derived the selection and adaption of another IA technique based on structural static and historical analysis. Another refinement was the development of the GBT estimation model. Due to this, HyEEASe comprised of two estimation paths, i.e., Hybrid estimation and GBT model estimation (C.4.1a and C.4.1b respectively as shown in Figure 4.7).

5.4.2 Evaluation object

HyEEASe was evaluated by using the prototype in a current running sprint at the case company for estimating the user stories.

The object of the evaluation was HyEEASe (i.e., including both Hybrid estimation and GBT model estimation paths). The Hybrid estimation path was evaluated for estimation effectiveness and perceived usefulness about the impact information of the potentially impacted code.

Regarding, the GBT model estimation path, it was also evaluated for estimation effectiveness. The model parameters were gathered by taking input from the experts at the case company. However, the output of the GBT model was not shown to them and evaluated afterward. It was done to avoid bias in experts' opinions during estimation.

5.4.3 Evaluation approach

This section describes the evaluation approach regarding goals, target population, design, execution, data collection, and analysis. The case study was conducted to provide a piece of empirical evidence that the HyEEASe method achieves its research goals when evaluated in a real industrial setting. In particular, all the three research goals G1, G2 and G3 were evaluated as seen in Figure 5.1.

Using the GQM template, the goal, therefore, was formulated as: "To understand and evaluate HyEEASe method supported by a prototype with respect to its usefulness and effectiveness from the perspective of agile development teams in the context of effort estimation in agile software development."

We evaluated the effectiveness of both the Hybrid estimation and GBT model estimation. The idea was to analyze whether using a purely expert-based (Planning Poker) or Hybrid estimation or a purely model-based (GBT model estimation) improves estimation accuracy. So basically, comparing both the estimation paths of HyEEASe to each other as well as to Planning Poker. We also wanted to identify needs for improvement to increase the quality of the method and prototype.

5.4.4 Evaluation criteria

To evaluate estimation reliability H1 and H2 (accuracy and bias) of the produced estimates, the following accuracy metrics [100] were considered:

- Mean Absolute Error (MAE), which is the average of the absolute errors between the actual and the predicted effort.

$$MAE = \sum_{i=1}^{TT} |AE_i - PE_i|$$

where AE_i = the actual effort collected from the dataset for the i^{th} test data, PE_i = the output (predicted effort) obtained using the developed model for the i^{th} test data and TT = total number of tasks (user story or change request) in the test set.

- Mean of Magnitude of Error Relative to the estimate (MMER) which is one of the criteria used for evaluating effort estimation models. It is shown that MMER can provide higher accuracy than Mean Magnitude of Relative Error (MMRE) [49] [66]. MMER is the mean of MER.

$$MMER = \frac{1}{TT} \sum_{i=1}^{TT} \frac{|AE_i - PE_i|}{PE_i}$$

- Prediction Accuracy PRED (x) which is the average of MAE's off less than or equal to x as shown in MAE. The accuracy of the estimates directly corresponds to PRED(x) and is conversely relative to MMER.

$$PRED(x) = \frac{1}{TT} \sum_{i=1}^{TT} \begin{cases} 1 & \text{if } MAE_i \leq x \\ 0 & \text{Otherwise} \end{cases}$$

These metrics were calculated for both the Hybrid estimation and the GBT model estimation paths (C.4.1a and C.4.1b in Figure 4.7 respectively) as soon as we obtained the actual efforts spent on each user story at the end of the sprint.

To evaluate H3 (usefulness) and H4 (learnability - awareness), several quality aspects such as usefulness, understandability, reliability, awareness, completeness, visualization, and acceptance were explored as

shown in Table 5.5. To quantify these aspects, a five-point Likert scale was used (scale: five-point ratio scale, with 5 being strongly agree and 1 being strongly disagree). The experts were asked to rate the quality aspects on this scale after they have used the prototype. Quality aspects rated with more than 3 (neutral opinion) show a positive evaluation overall.

5.4.5 Evaluation design - sample and population

The evaluation was conducted in a case study setting with Insiders Technologies.

The corresponding agile development team at Insiders Technologies was contacted with whom the data for the method were collected because they were the right stakeholders and also the end users of the method and tool. Therefore, their feedback is very relevant. Four participants from the team with various roles were able to participate. One of these participants was both the Scrum Master and a developer; the rest were developers. The aggregated experience of all the participants was 3.5 years (median value) in agile development and 2.5 years in estimation. Regarding the context, domain, type of product, etc. please refer to Section 2.4.2.

At the case company, a user story is the highest level of granularity, which is initially estimated roughly. Later, these stories are decomposed into smaller “Tasks”, which are then estimated in a Planning Poker meeting and finally implemented.

Since the planned total number of tasks (i.e., new features and not defects) for the running sprint were only two, this motivated us to include other already implemented tasks for our evaluation. These tasks were chosen randomly from the database together with our contact person other than the participants. Associated threats are mentioned in Section 5.4.10. While choosing the tasks, we made sure that they were estimated and/or implemented way back in time and with developers other than the ones participating in the evaluation. It was done to reduce any bias and/or participant’s familiarity with the estimated or actual effort. Keeping the duration of the evaluation in mind, two such tasks were selected making a total of four. One was overestimated, and the other was just right. We wanted to analyze how our HyEEASe would perform compared to the initial purely expert-based estimation.

The protocol used in execution includes (1) the procedures for inviting the participants and making arrangements for the fieldwork, (2) the informed consent form allowing us to collect data for the intended analysis and informing the participants about the evaluation as well as their rights, (3) the data collection procedures, and (4) the data analysis procedures.

5.4.6 Organizational context

The case company also employs the “Clean Code” policy [77]. According to this policy, no comments or identifiers are added in the code which meant textual analysis could not be applied for performing impact analysis. Furthermore, based on the argumentation in Section 4.3, a combination of historical and structural static impact analysis technique were selected, adapted and applied to perform IA.

As explained earlier that at the case company, a user story is the highest level of granularity, which is then decomposed into smaller tasks. These tasks are estimated and implemented. It is for these tasks that impact analysis has been performed. The intention was that the experts use the prototype during Planning Poker and, by consensus, estimate the effort for a given set of tasks. The case company uses a multistage process to estimate a user story as described in Section 2.4.7. HyEEASe was applied in the Q&A stage in Section 2.4.7 where the tasks got their final estimates during Planning Poker.

5.4.7 Execution and data collection

We conducted an interactive workshop where we first introduced the prototype and later the participants used it themselves for estimating efforts for the tasks. To reduce bias and increase the credibility of the empirical data and findings, multiple data sources were used. These sources include observation of the effort estimation process in the evaluation workshop, the questionnaire where the participants rated the quality aspects of the prototype and the moderation cards where the participants provided their open feedback. The execution steps are as follows:

1. Introduction: We explained the goals and procedure of evaluation and collected the signed informed consent forms.
2. Training: We introduced the idea of using change impact analysis for effort estimation and demonstrated the prototype by explaining its functionality and resolved any open questions.
3. Task estimation: In this part of the evaluation, we asked the participants to use the prototype to estimate the effort for four tasks. After considering the information provided by the prototype, they estimated the effort for each task. The estimations were stored using the prototype interface. If they had different opinions, they discussed their estimated efforts and re-estimated them until they agreed. During this activity, we observed the participants, took notes and answered questions. After considering the information provided by the prototype, they estimated the effort for each task.

- Using Hybrid estimation, the experts estimated the effort for each task. If they had different opinions, they discussed their estimated efforts and re-estimated them until they agreed. During this activity, the participants were observed, notes were taken, and questions were addressed.
 - After using Hybrid estimation, the experts provided input to the parameters of the GBT model. The estimates produced by the GBT model were not shown to the experts to keep their estimates (made through using the prototype) unbiased.
4. Feedback: After the estimation, the experts were provided with a questionnaire to rate several quality aspects as seen in Table 5.5. To collect their feedback, we then performed another interactive session. We asked the participants to provide their reflections (both strengths and improvement suggestions) using moderation cards. The cards were then collected, discussed and clustered accordingly.

This evaluation workshop lasted four hours. During the whole evaluation, two researchers were present; one was the moderator, and the other one was acting as an observer and taking notes. The estimates made through using the Hybrid estimation and the input to GBT model parameters were collected through the prototype. The quantitative data, i.e., the rating of the quality aspects, was collected using a questionnaire that was designed using TAM[116] and the instrument provided by McKinney et al. [78]. The qualitative data, i.e., the feedback, was gathered through moderation cards where the participants provided their open feedback. The materials can be found in Appendix D.

5.4.8 Data analysis

The input provided by the experts to the parameters of the GBT model was then analyzed. In case of any missing parameters, the impact analysis database was consulted to compensate for the missing values. The model was then applied to this new task dataset offline, and estimates were obtained. The quantitative and the qualitative data were analyzed, compared and results were integrated. The quantitative analysis was done by using the evaluation criteria described in 5.4.4. For the qualitative data, themes were derived from the feedback session.

5.4.9 Results and interpretation

This section presents the results of the evaluation and discusses them simultaneously.

1. Results of evaluating H1 and H2:

To evaluate the hypotheses H1 and H2, the effort estimated by the experts using Hybrid estimation as well as the effort estimated by the GBT model (by applying it offline) were analyzed and compared with the actual effort data. The results of this comparison are reported using MAE, MMER, and Pred metrics as shown in Table 5.4.

Table 5.4: Comparison of estimation accuracy

	Purely expert-based effort estimation (Planning Poker)	Expert-based effort estimation using Hybrid estimation	Purely model-based estimation using GBT model
MAE	0.72	0.22	0.88
MMER	1.55	0.23	1.03
Pred(25)	0.5	0.75	0.5

MAE and MMER are negatively oriented scores means the low the value the higher accuracy is. A method with low MAE, MMER and high Pred gives more accurate results.

It is clear from the table that MMER and Pred of expert-based estimation using Hybrid estimation is better than both purely expert-based (Planning Poker) and purely model-based estimation (GBT model). Also for the already implemented overestimated task, the estimates were improved (getting closer to actual effort) when re-estimated using Hybrid estimation. Furthermore, upon sharing the results with the participants afterwards, we were informed that for one of the new planned tasks (T1 in Figure 5.6) for the current sprint, the team revised their initial purely expert-based estimates (from 1.0 to 1.5 PD) in their Q&A session, based on the impact information they got from our evaluation. During the evaluation, they had estimated T1 as 2.0 PD using Hybrid estimation where the actual turned out to be 2.3 PD.

This indicates that Hybrid estimation enabled the experts to do better estimations by supplying all relevant and necessary impact information and led to improved accuracy of estimates than the purely expert-based estimation (Planning Poker).

On the other hand, the performance of the GBT model for estimating tasks was not good. One possible reason could be the missing data. The experts could only supply values for the LOC parameter. The missing parameters were provided by taking mean values from our database. This mechanism of providing missing information was perhaps not optimal, hence resulting in the low performance of the model. The comparison of estimated effort with Planning Poker, with Hybrid estimation, with the GBT model and with the actual effort is shown in Figure 5.6. From the figure and the above-mentioned results, we can conclude that neither the purely expert-

based (Planning Poker) nor the pure model-based estimation (GBT model) improved estimation accuracy. However, expert-based estimation using Hybrid estimation support managed to improve estimation accuracy.

These results are in line with the results found in the literature concerning model-based estimation methods and their performance in terms of improving estimation. It is established [62] that it is not possible to conclude whether model-based or expert-based models perform better. However, expert estimates seem to be more accurate if domain knowledge is not included in the model or when uncertainty is high. The results also align with our previous case study [106] in which we found that to improve expert-based estimation, decision support is required that provides useful historical data and impact information.

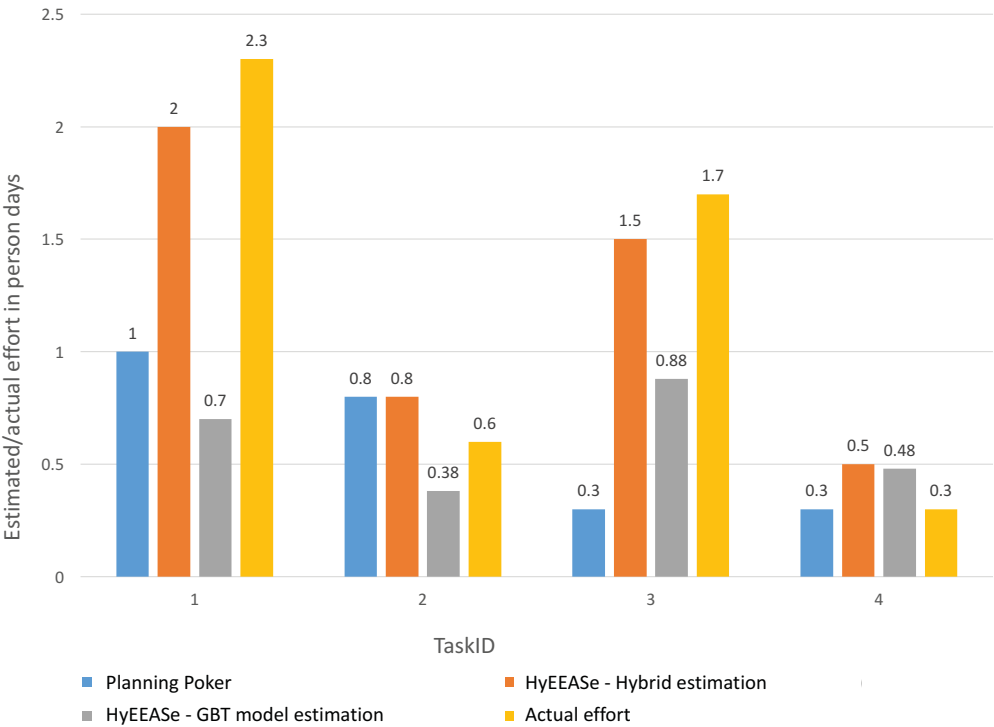


Figure 5.6: Comparison of estimation accuracy - Pure Expert-based (Planning Poker) vs Expert-based using Hybrid estimation vs Pure model-based (GBT model estimation)

- 2. Results of evaluating H3 and H4: Hypotheses H3 and H4 were evaluated, and the results are shown in Table 5.5. The practitioners were asked about their level of agreement regarding the quality aspects on a five-point Likert scale.

Table 5.5: Results of the evaluated quality aspects of the prototype

Quality aspect	Reference	Result (median)
Usefulness	McKinney[78]	4
Reliability	McKinney[78]	3.75
Relevance	Lee and Strong[68]	3.25
Visualization	Nelson et al.[86]	3
Acceptance	TAM[116]	3
Completeness	Goodhue et al.[51]	2

The values are median values where $N = 4$. Overall, the participants found the prototype useful, reliable, and relevant for estimation. However, regarding its Acceptance, the participants were neutral (median = 3). This might be because during the feedback session they stated that if the developer is unfamiliar with the underlying system and the prototype, it might be difficult to apply it (for details, see improvement suggestions in Table 5.3).

Moreover, in their view, the concept does not take into account certain code metrics such as Modified LOC or Test metrics, which made them disagree on its Completeness (median = 2). Here we must emphasize the constraint that the underlying static code analyzer does not provide these metrics; therefore we do not have them in this prototype. Nonetheless, we intend to compensate for this lack in our next development iteration.

The summary of the positive feedback with categories is shown in Table 5.2. In general, the participants appreciated the availability and accessibility of the impact and relevant historical information. They found the prototype useful to support estimation when experts do not agree on a particular estimation. The prototype also visualizes the impact information and shows past estimation trends (over-/under-estimation). Impact information visualization helps them to learn which other parts of the system they need to consider while estimating.

A summary of the improvement suggestions, their categories, and explanations by the practitioners along with the response from the author addressing the corresponding suggestion are presented in Table 5.3. The participants suggested that in addition to impact information, it could be useful for estimation if a link to code changes was provided. We appreciate the suggestion; however, as the case company has a clean code policy, i.e., no commits/identifiers are added in the source code, it would be difficult to establish such links without this information. However, a workaround is possible. Furthermore, the practitioners had concerns regarding missing metrics in the prototype. For example, they raised the importance of having the code metric Modified LOC. The concerns are valid; however, it has been explained already that these missing metrics are limita-

tions because the underlying static code analyzer does not provide these metrics.

Table 5.6: Positive feedback given by practitioners

Category	Feedback
Visualization	Several kinds of visualization are available regarding the impact set (impact classes). Code structural dependencies are visible through dependency graphs. Visibility of impact. Dependencies of impact classes on other classes are visualized (through charts/tables).
Estimation support	Assists in implementing a change during estimation. Shows errors regarding the estimated effort of the related tasks implemented in the past.
Learning	The prototype enables learning and is useful especially if the experts do not agree on the estimated effort for a backlog item.
Impact granularity and scope	Shows the potential impact at the class level. Enables one to think about which classes may be impacted before implementing the backlog item.
Access to (historical) data	Availability of huge amount of historical data regarding related tasks and classes. Shows impact factor data on the related tasks and classes. Easy access to past data. Data accessibility on different granularity levels, i.e., tasks, class.

5.4.10 Threats to validity

This section discusses some threats to the validity of this evaluation and how they were mitigated.

- **Construct validity:** To avoid inappropriate measurement of the evaluation quality aspects, existing reliable test instruments such as TAM [116] were considered. Based on peer reviews, and experiences gathered in the study [108], the appropriateness of the questions for the evaluation goals was checked. Please refer to Section 5.3.9 for a detailed discussion of threats to construct validity of this study.
- **Internal validity:** To reduce the bias introduced by participant's familiarity with the estimated or actual effort for the two already implemented tasks in the dataset, we made sure that the tasks were randomly selected, were estimated or implemented way back in time. Additionally, developers other than the ones participating in the evaluation implemented these tasks. As pointed out in the improvement suggestions by practitioners (see Table 5.7) not con-

Table 5.7: Improvement suggestions given by practitioners and corresponding response from authors

Suggestions by practitioners	Response by authors
General suggestions	
In addition to impact data, links to the changed code of old backlog items may also be useful.	Due to the clean code policy at the case company, there are no comments/identifiers in the code. Therefore, no direct way exists for tracking and linking the changed code with the backlog item. Nonetheless, some workaround could be found in future work, such as tracking the changed lines of code in the version repository against particular commits and user stories against every associated task and every affected class.
Missing features	
Other metrics: e.g., Modified LOC, testing metrics are missing.	The Modified LOC is rather a constraint instead as the underlying code analyzer does not support it. However, a workaround may be possible in the future to compensate for these missing metrics.
Constraints	
Expert familiarity and knowledge: Expert familiarity and knowledge (for prototype and data usage) necessary for increasing productivity.	To use the prototype, training is required. However, in the prototype, additional help via menus/files can be supported additionally to increase understandability. The upfront investment is required to understand the prototype to increase productivity (a cost-benefit trade-off).

sidering relevant factors like Modified LOC and testing metrics also poses a risk to the validity of the results.

- **External validity:** As the evaluation used a convenience and small sample, the results are not statistically generalizable. To increase the representativeness of the results, the study could be replicated in other cases. Similarly, to increase the representativeness of our results within the company, we will need to use the method for multiple sprints in the company.

However, we can argue that the study has theoretical generalizability, as similar results can be expected in a similar context with similar characteristics of the participants for agile development. Given that the most commonly used effort estimation method in the industry is Planning Poker and that most companies have version management and issue tracking systems, the results will likely be applicable in most cases. The use of existing data from various information silos also reduces the cost of the intervention for any adopting company [18].

More replications in different companies will also improve the generalizability of the results. The general instance of HyEEASe, however, will need to be operationalized in each case company. The main limitation of the applicability of the method proposed in this study is that it will not apply to completely new development projects.

- **Conclusion validity:** The results of the participants were aggregated, and only median values and the number of practitioners were used to identify common practices/views and to point out potential future research areas. However, due to the small sample, more evaluation studies are needed to increase statistical power. To improve the confidence in the interpretation of the data and validity of the conclusions, we took several steps e.g. a second researcher reviewed the analysis and the results and our interpretations were also presented to the companies for validation.

5.4.11 Conclusion

We have reported the results of case study research evaluating the hybrid effort estimation method supported by a prototype for estimation effectiveness and several other quality aspects. It is concluded that practitioners perceived the overall method and prototype as very useful for supporting EE. The impact information and the visualizations not only supported them during estimation but also enabled them to learn about the system. Furthermore, in their view, considering modified LOC and testing metrics among the impact factors could improve the effectiveness of IA for EE. Hence, more metrics are required to be explored for a more suitable method. Regarding estimation accuracy, HyEEASe - GBT model did not perform better than HyEEASe - Hybrid estimation as the later provided additional useful impact information to the experts.

5.5 Data-based evaluation

In this data-based evaluation, HyEEASE (GBT model estimation only) was also evaluated for the reliability of produced estimates by comparing it with Agile COCOMO II [33]. This evaluation was also done in close collaboration with Insiders Technologies. The data (GBT model parameters) gathered during the case study (see Section 5.4) was reused and compared with Agile COCOMO II. The setup of Agile COCOMO II was also done by the scrum master of the team we did the evaluation with (see Section 5.4).

Agile COCOMO II incorporates the full COCOMO parametric model and uses analogy-based estimation to generate accurate results for a new project. The analogy is based on already completed projects and only changes in COCOMO parameters [1]. In the given context of Insiders Technologies, the new project is a new task (or change request or backlog item), where completed projects refer to the already implemented tasks.

5.5.1 Evaluation goals and hypotheses

With this evaluation, the performance of the GBT model and Agile CO-COMO II were compared regarding the reliability of produced estimates and provided another evidence that the proposed method achieves its research goals when evaluated in comparison to another research alternative. G1 was evaluated as seen in Table 5.8.

Table 5.8: Evaluated research goal and hypotheses

Research goal	Hypotheses
G1: To increase the reliability	H1: GBT model increases the estimation accuracy. H2: GBT model reduces estimation bias.

5.5.2 Evaluation criteria

To evaluate reliability of the produced estimates, the same evaluation criteria was considered as seen in Section 5.4.4 i.e. MAE, MMER and Pred (x).

These metrics were calculated for the estimates estimated by both the GBT model and Agile COCOMO II as soon as the actual efforts spent on each task were obtained at the end of the sprint. These metrics were compared to each other and to the baseline, i.e., the estimated effort by Planning Poker as shown in Table 5.4. Before explaining the set up of Agile COCOMO II parameters for the comparison, the model itself is briefly described below.

5.5.3 Agile COCOMO II

The objective of the Agile COCOMO II estimation model was to provide support for project planning and scheduling, project staffing, estimates-to-complete, project replanning/ rescheduling, project tracking, contract negotiation, proposal evaluation, concept exploration, design evaluation, and bid/no-bid decisions [33]. COCOMO II provides the following sub-models for estimation of software projects [33]:

1. Application Composition model: It is used when software is composed of existing parts. The earliest phases or spiral cycles will generally involve prototyping, using this model.
2. Early Design model: The next phases or spiral cycles will generally involve exploration of architectural alternatives or incremental development strategies, i.e., the model is used when requirements are available, but the design has not yet started (has six cost drivers).

3. Reuse model. It is used to compute the effort of integrating reusable components.
4. Post-Architecture model: It is used once the system architecture has been designed and more information about the system is available (has 17 cost drivers).

It is indicated [33] that, COCOMO II uses the reuse model for maintenance when the amount of added or changed base source code is less than or equal to 20% or the new code being developed [8]. The base code is source code that already exists and is being changed for use in the current project.

Given the assumptions/premise of the method (see Section 4.9), i.e.

- The system or software to which the change requests (user stories or backlog items) are being made is partially developed means the base code exists, and functionality is being added to it incrementally in each sprint.
- The change requests are not completely independent of the existing system.

The best-suited option was, to compare the GBT model with the “COCOMO II Reuse model” for maintenance in terms of the effectiveness of estimates. The details on the COCOMO II Reuse model can be found in Appendix E.

5.5.4 Set up of Agile COCOMO II

As explained in the previous Section 5.4.7, the team had four tasks to do the estimation using the hybrid method in the planning meeting. It was required to set up the model parameters, the cost drivers and the scale factors associated with Agile COCOMO II for these four tasks.

A list of all the scale factors and the cost drivers with their possible ratings were provided to the scrum master of the agile development team with whom the hybrid method was evaluated. The scrum master could analyze and rate the cost drivers that will be changing across each task and by how much. The scale factors were rated once for all the tasks since they were considered to stay constant over a sprint. Table E.1 in Appendix E shows the complete list of scale factors and Table E.2 in Appendix E shows the cost drivers and their corresponding rating by the scrum master.

5.5.5 Results and comparison

With the SizeAdded and SizeModified parameters and the scale factors and cost drivers, the effort was estimated for each of the four tasks

using the Agile COCOMO II model (see Table 5.9 estimated effort of each task in person-months (PM) and person-days (PD)). COCOMO II treats 1 PM as 152 person-hours means 19 PD excluding weekends, holidays and vacations.

Table 5.9: Estimated effort for the tasks using Agile COCOMO II

	Task 1	Task 2	Task 3	Task 4
SizeAdded	10	30	50	20
SizeModified	30	20	40	20
MAF	1.18	1.18	1.12	1.06
Size	47.20	59.00	100.80	42.40
Constant A = 2.94				
Constant B = 0.91				
$E = B + 0.01 * \text{Sum of Scale factors (13.87)} = 1.05$				
PM	2495.49	3081.27	5401.66	2160.02
PD	47414.39	58544.19	102631.52	41040.40

To evaluate the reliability of the estimates, the effort estimated by the experts using Hybrid estimation, the effort estimated by the GBT model as well as the estimates generated by Agile COCOMO II were analyzed and compared it with the actual effort data. The results of this comparison using MAE, MMER, and Pred metrics are shown in Table 5.10. It is clear from the table that MMER and Pred of Hybrid estimation are better than Planning Poker, GBT model and Agile COCOMO II. However, the results of the GBT model are notably better (both in terms of accuracy and bias) than Agile COCOMO II.

Table 5.10: Comparison of estimation accuracy - H1

	Purely expert-based effort estimation (Planning Poker)	Expert-based effort estimation using Hybrid estimation	Purely model-based estimation using GBT model	Agile COCOMO II
MAE	0.72	0.22	0.88	3283.3
MMER	1.55	0.23	1.03	0.99
Pred(25)	0.5	0.75	0.5	0

5.5.6 Conclusion

This section reports the evaluation of the GBT model by comparing it with Agile COCOMO II regarding estimation reliability. Out of all the model versions, the reuse model of Agile COCOMO II was used as it was the closest match to the GBT model in the given context. The required

model parameters, the scale factors, and the cost drivers were rated by the scrum master of the case company. The effort estimated by the GBT model as well as the estimates generated by Agile COCOMO II were analyzed and compared with the actual effort data. The results of this comparison are reported using MAE, MMER, and Pred metrics as shown in Table 5.10. Table 5.10 shows a magnitude of difference in accuracy and bias of estimates generated by the GBT model as compared to Agile COCOMO II. It is therefore concluded that the GBT model generates notably accurate and notably less biased estimates as compared to Agile COCOMO II.

5.6 Controlled experiment with students

This section reports a controlled experiment that was conducted with the BS and MS students taking the Software Process and Project Management (SPPM) course at the Department of Computer Science, TUKL. The experiment is reported following the guidelines as suggested by Wohlin et al. [118] and Jedlitschka et al. [61].

5.6.1 Motivation and context selection

After conducting the case studies, the results of evaluated hypotheses, i.e., usefulness(H3 - regarding understandability and completeness) and learnability (H4) were compared among the case studies. The comparison in Table 5.11 showed an almost equal response with respect to understandability whereas the completeness and learnability could not be compared since they were evaluated only in one case study with Insiders Technologies. This is because case study research with SAP was done using a mock-up with which completeness and learnability could not be evaluated. Therefore, to investigate further deep and understand if there would be differences among these quality attributes (H3, H4), the experiment was conducted. To achieve the most general results in an experiment, it should be executed in large, real software projects, with professional staff [118]. However, conducting an experiment in a real-life setting would incur significant costs, may involve risks and require a lot of time. Therefore, a cheaper alternative was chosen, i.e., conducting an experiment with students as subjects. The experiment was done with a mixture of BS and MS students taking the SPPM course at TUKL. Altogether 25 students participated. The study is a controlled experiment.

The experiment can be considered as general in the sense that the objective is to compare and evaluate two estimation methods in general (from a research perspective), and it is not about comparing an existing estimation method in a company with a new alternative estimation method [118].

Table 5.11: Comparison of results of hypotheses across case studies

Evaluated Hypotheses	Case study at SAP (median values)	Case study at Insiders Technologies (median values)
H3 - Understandability	4	3.5
H3 - Completeness	NA	2
H4 - Learnability	NA	2.75

5.6.2 Experiment design

In this subsection, the goals, hypotheses, evaluation criteria, variables, subjects and objects, design, instrumentation, and evaluation of validity are described.

1. Goal: The goal is to determine the differences between two estimation methods, i.e., Planning Poker versus HyEEASe (Hybrid estimation only) with respect to H3: usefulness and H4: learnability hypotheses.
 - Object of study: The objects of study are the estimation methods.
 - Subjects: The subjects are the students in the SPPM course.
 - Purpose: The purpose of the experiment is to compare and evaluate the estimation methods.
 - Perspective: The perspective is from the point of BS and MS students (novice developers) based on their background and experience with software development and estimation.
 - Quality focus: The quality focus is the usefulness and learnability of the estimation methods.

To summarize the goal based on GQM template: "Analyze Planning Poker and HyEEASe estimation methods for the purpose of comparison with respect to the usefulness and learnability from the point of view of BS and MS students (novice developers) in the context of SPPM lecture."

2. Hypotheses:

As stated previously, the intention was to compare and evaluate both H3: usefulness and H4: learnability when using two estimation methods for doing the estimation. The null hypotheses are:

- $H3_0$: HyEEASe provides equally useful (understandable, complete) impact information of the potentially impacted code as compared to Planning Poker.

- H4₀: HyEEASe enables equal learning (awareness) of the impact factors as compared to Planning Poker.

The alternative hypotheses are:

- H3₁: HyEEASe provides more useful (understandable, complete) impact information of the potentially impacted code as compared to Planning Poker.
- H4₁: HyEEASe enables more learning (awareness) of the impact factors as compared to Planning Poker.

Formally,

- H3₀ and H4₀ : HyEEASe $(Usefulness, Learnability)$ = Planning Poker $(Usefulness, Learnability)$
- H3₁ and H4₁: HyEEASe $(Usefulness, Learnability)$ > Planning Poker $(Usefulness, Learnability)$

3. Evaluation criteria: The experiment had certain tasks to be performed. The responses to these tasks were used to evaluate the hypotheses. Table 5.12 provides a complete picture of the tasks performed by the students during the experiment and their mapping to the evaluated hypotheses along with evaluation criteria.
4. Variables: The independent variable is the estimation method at a non-fixed level and experience of the students (for software development and estimation) at a fixed level. The dependent variables are the usefulness and learnability. Usefulness has two dimensions understandability and completeness. Table 5.12 provides the details on how these variables are evaluated.
5. Subjects and objects: The selection of subjects was based on convenience, i.e., BS and MS students participating in the SPPM course at the university. The students had the freedom to refuse participation, without any penalty for the individual. However, to encourage participation students were promised 5% bonus on their score in the final exam. The registration also had a pre-screening test questionnaire to judge the background and experience of the students with software development and estimation. With this test, the subjects were divided into experience/inexperience groups that helped decide the design. To ensure equal treatment (right to service), in a follow-up seminar the first author presented both methods to students in the course. All students were welcome to attend the lecture regardless of their participation in the experiment.

The objects were the set of tasks that were to be performed during the experiment.

6. Design: The chosen design type was one factor with two treatments. The factor in this experiment is the estimation method, and

Table 5.12: Mapping of tasks to evaluated hypotheses

Evaluated hypotheses	Task No. and description	Explanation (expected responses)	Criteria
H3: Understandability	T1: Assessment of statements as right/wrong about the estimation method.	If the students could rightly identify correct statements that would show they understood estimation method.	The no. of students answering correctly.
H3: Completeness	T2: Assessment of provided information for doing effort estimation (information that may influence estimation).	If the students would only choose from the provided information and did not provide any additional information, it would show the completeness of estimation method.	The no. of unique additional information per student. If the students provided many items, the method is less complete in comparison to providing fewer items.
H4: Learnability	T3: Assessment of the provided effort after considering the provided information (with rationale).	If the students would change their effort estimates, after considering the provided information, it would show that they have learned about the impact factors and their influence on estimation. If the students in their rationale, mention any of the provided information, it would also show they learned about the factors.	The no. of students mentioning any of the provided information (impact factors) while assessing the estimated effort.
H3 and H4	T4: Perception of usefulness and learnability.	This was captured using TAM concepts through a feedback questionnaire.	Median values more than 3 for H3 and H4.

the treatments are Planning Poker and HyEEASe (Hybrid estimation only). Based on the context, a completely randomized design was chosen. The design setup used the same objects (set of tasks) for both treatments and assigns the subjects randomly to each treatment (estimation method). Each subject used only one treatment on the object [118].

7. Instrumentation: A pre-screening test was developed to find the background and experience of the individuals. Introduction to the experiment and the link to the pre-screening test was provided to the students at the first lecture (see Appendix F). This data provided the input to the characterization of the students. The guidelines for using both the estimation methods were developed using a fictitious example scenario based on a real system (i.e., Moodle). Data collection forms were also developed. The materials can be found in Appendix F.

Table 5.13 shows the experiment design.

Table 5.13: Experiment design in detail

Parameter	Explanation
Independent variable	Estimation method (non-fixed level), Experience of the students (fixed level)
Dependent variable	Perception on: 1) Usefulness (Understandability, Completeness) -> H3 2) Learnability (Awareness)-> H4
Subjects	BS and MS students taking the SPPM course
Sampling	Stratified, random
Design	One factor two treatment
Factor	Estimation method
Treatments	Planning Poker and HyEEASe estimation methods
Task to be performed	1. Understanding the estimation methods. 2. Analyzing effort estimates of enhancement requests and factors influencing estimates
Rights to service	After the experiments, the students were presented with both estimations methods

5.6.3 Execution

In this subsection, the execution of the experiment is described.

Sample

As mentioned earlier, the background and experience of the students were found through a pre-screening test. The sample was stratified, i.e., based on the outcome of the test, the subjects were divided into experience/inexperience groups from which subjects were randomly assigned to the groups in the experiment. This was done to ensure that the groups were as equal as possible while maintaining the randomization over the subjects. It was also intended to have a balanced design, i.e., having an equal number of students in each group but due to an unequal number of students in experienced/inexperienced strata, it was 13 students in the controlled group and 12 in the treatment group thus an imbalanced design. Each group had a mix of experienced and inexperienced students making them comparable. Table 5.14 shows the details of each group.

Preparation

It was made sure that the required infrastructure was in place. This included booking the room with sufficient seats and space to place both groups in. Then the computer and beamer to introduce the process of

Table 5.14: Experiment group details

Details	Group A	Group B
No. of experienced students	6	5
No. of inexperienced students	7	7
Total no. of students	13	12
Treatment	Planning Poker	HyEEASe

experiment execution. The subjects were not aware of the actual hypotheses stated. They were informed that the researchers wanted to study the outcome of the comparison of different estimation methods based on the evaluation done by the subjects. The experiment material was prepared in advance. Preparation materials can be found at Appendix F. It included a booklet for every student, separate forms with a description of tasks to be performed with related questions and a feedback questionnaire. The contents of the booklet were as follows:

- Introduction to the software effort estimation
- Introduction to the context (i.e., the system about which the tasks were to be performed). The open source system, "Moodle"¹ was chosen since students mostly are familiar with it and at TUKL they have been using some similar systems in their daily routine of courses registration, assignment submission, etc. The system was briefly introduced, its architecture and its main features.
- Introduction to the enhancement required for one of the main features, i.e., Plagiarism detection.
- The scenario (a profile of a fictitious company and a profile of the reader, i.e., the subjects. The subjects were said to be the employees of the fictitious company).
- Walkthrough of Planning Poker and HyEEASe estimation methods for controlled and treatment groups respectively.
- The set of tasks to be performed by the subjects.

Copies of the booklet, tasks, feedback questionnaire and data collection forms were made available for all subjects in both the groups.

Execution

The experiment was executed on April 30th, 2018 during the lecture class of SPPM for about 1.5-hours duration. A quick introduction to the experiment format was given at the start. Both groups, as well as each student, was suitably spread out in the room. Both groups performed the same set of tasks at the same time using two different estimation

¹Moodle: <https://moodle.org/>

methods. After completion of one task, the students were required to hand over their completed tasks and then request for the next. The experiment ended with the handing over of the feedback questionnaire. A helper was also there to help the researcher with the handling of groups. The researcher was there throughout the experiment to help answer any question. Figure 5.7 shows the overall execution process of the experiment.

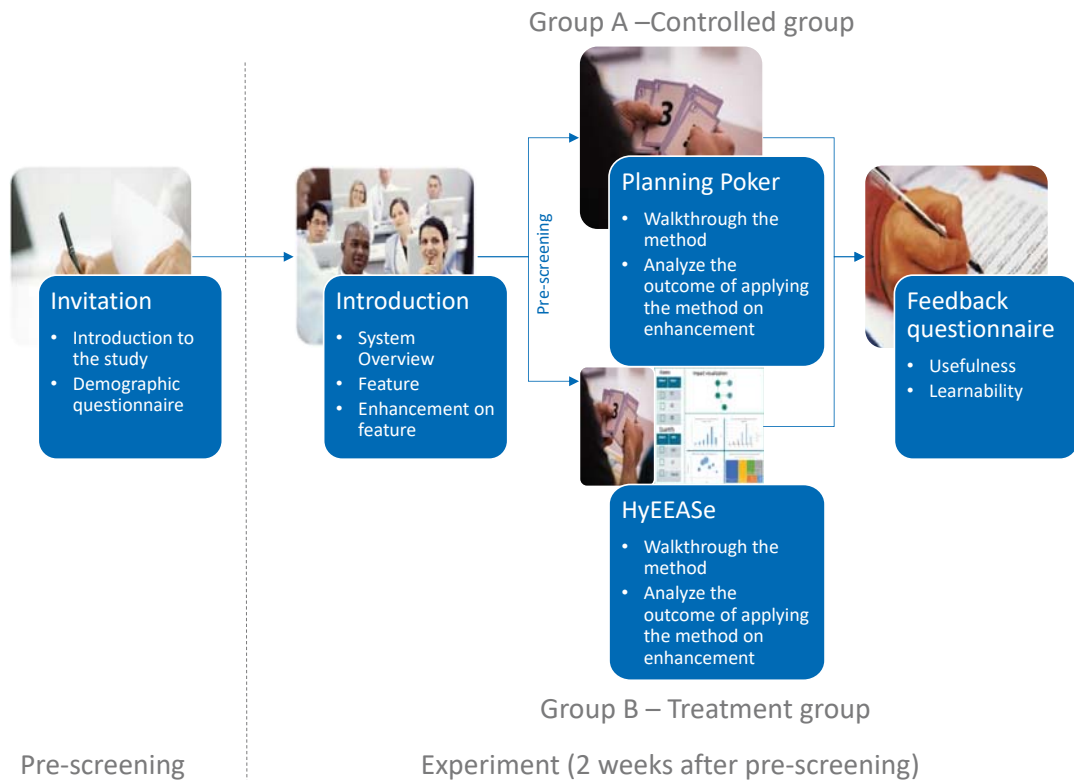


Figure 5.7: Experiment execution

Data validation

Finally, the data was validated. Before a subject would leave the experiment upon finishing, his/her data form was checked to ensure that they have filled in the forms. Data were collected for 25 students, 13 for controlled and 12 for treatment groups. However, one student in the treatment group was removed since the data was incomplete, leaving 11 students in that group.

5.6.4 Threats to validity

This section discusses some threats to the validity of this evaluation and ways to mitigate them.

- **Construct validity:** To avoid an inadequate operation, the experiment was operationalized using the GQM template. To avoid inappropriate measurement of the evaluation quality aspects, existing reliable test instruments such as TAM [116] was considered and adapted. Based on peer reviews and experiences gathered in our study [108]; we checked the appropriateness of the questions for our evaluation goals. A relevant questionnaire was posed to students to elicit their feedback about the estimation methods. The mono-operation bias (as there were only four tasks to perform) could not be avoided due to time constraints. To avoid mono-method bias, quantitative and qualitative analysis was combined and compared. We did not pilot the execution of the experiment with a small set of potential participants due to practical constraints. However, by involving more than one researcher in each step of the experiment design, review, execution and the use of the defined protocol, the reliability of the experiment was increased.
- **Internal validity:** There could be selection bias as the sample was convenient. We however assigned the students different estimation methods using stratified random sampling, where strata were defined based on their self-assessment of knowledge and experience in agile software development and software effort estimation. To reduce the impact of unfamiliarity with the software system, i.e., regarding the instrumentation, we presented the students with an open source system, and features and enhancement requests related to it. Regarding the instrumentation, the students were presented with a generic open source system with features and enhancement requests related to it.

To avoid evaluation apprehension, the students were assured of the anonymity of their personal and organizational data.

Furthermore, experimenter bias was reduced by following the predefined guidelines for the experiment.

The experiment was not piloted with any other subjects before running it with students and thus is also a threat to validity.

- **External validity:** Convenience sampling and small sample size for the evaluation with student reduces the statistical generalizability of our results. However, given the consistent positive evaluation from the students participating in the study gives confidence that the method is understandable. To avoid selection treatment interactions replications are required. To mitigate setting treatment interaction, the tasks were related to a real open source system.
- **Conclusion validity:** The results of the students are aggregated, and only median values and the number of students to identify common views are used. However, due to the small sample, more evaluation studies are needed to increase statistical power.

Before running the experiment, few colleagues at Fraunhofer IESE who are experts in conducting empirical studies reviewed and helped improve the design of the experiment. To motivate students to participate in the experimental study all participating students were promised extra points in the form of 5% of their score in the final exam. They were further encouraged to participate since they can learn about state-of-the-art effort estimation methods. The extrinsic incentive was fairly small, so the students' participation was voluntary, as can be seen, that a few students did not sign-up for the study. Furthermore, by concealing the goal of the study, by promising confidentiality of responses and by disassociating the incentive of participating from the performance in the experiment we increased the validity of the study.

5.6.5 Analysis and interpretation

This section presents the results of the hypotheses (i.e., H3 and H4) testing and discusses them simultaneously.

The number of students in both the groups, i.e., 13 and 11 respectively was very limited; therefore non-parametric tests like Mann-Whitney U could only be performed, and the results were explained using boxplots and histograms for testing different hypotheses. Due to the small sample, no significant result was obtained.

The figure 5.12 gives an overview of the mapping of hypotheses and the tasks. It is used below to explain how the hypotheses were tested.

1. H3 - Usefulness (Understandability): The hypothesis regarding the understandability of the estimation method was tested by performing T1 in Table 5.12. The criteria to measure understandability was the number of students answering correctly. Figure 5.8 shows the results of performing the task. Students in the treatment group, i.e., HyEEASe, provided more right answers indicating they have understood the method.
2. H3 - Usefulness (Completeness): The hypothesis regarding the completeness of the estimation method was tested by performing T2 in Table 5.12. The criteria to measure completeness was the number of students giving additional information/ impact factors other than the provided information that influences estimation. The provided information (as options) was as follows: developer's knowledge, developer's development experience, developer's estimation experience, complexity of described feature and the system, number of potentially impacted components, size of the potentially impacted components, complexity of the potentially impacted components, dependencies among the potentially impacted compo-

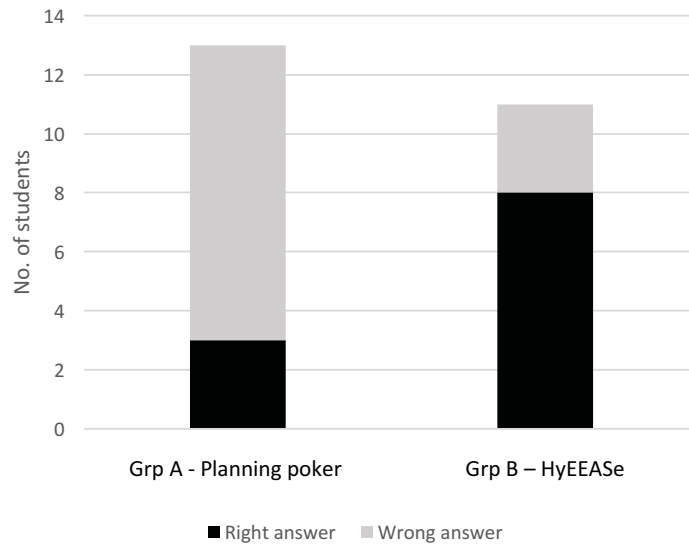


Figure 5.8: Testing the H3 - Usefulness (understandability) of estimation methods across both the groups

nents, effort estimates made in the past for the same impacted components, others.

Two students out of 13 in the controlled group gave information in addition to selecting from the provided information as "others," i.e. "workload of the team" and "knowledge of tools/technology".

Whereas only one out of 11 students in the treatment group gave information in addition to selecting from the provided information as "other," i.e. "number of required testers". Students in the treatment group provided less additional information compared to the controlled group which indicates they found HyEEASE more complete as compared to Planning Poker.

The results indicate that the alternative hypothesis $H3_1$ could be accepted.

3. H4 - Learnability: The hypothesis regarding the learnability of the estimation method was tested by performing T3 in Table 5.12. The criteria to measure learnability was the number of students mentioning any of the impact factors while assessing the estimated effort. Figure 5.9 shows the results of the learnability of the estimation method across the controlled and treatment groups.

Both groups were first asked to assess an estimate of a feature (which was estimated as 8 story points in this task description, (see Figure, i.e., 5.9 in A) without providing the information (i.e., impact factors as seen in the previous task).

The results of the controlled group were as follows (when no impact information was provided to them):

- Eight students out of 13, agreed to the estimate.
- One of the students did not agree to the estimate.
- Whereas four of the students did not have any opinion as they replied "I don't know".

The results of the treatment group were as follows (when no impact information was provided to them):

- Eight students out of 11, agreed to the estimate when no information was provided to them.
- One of the students did not agree to the estimate.
- Whereas two of the students did not have any opinion as they replied "I don't know".

After this, both groups were asked to re-assess the estimate of a feature which was estimated as 8 story points in this task description, (see Figure 5.9 in B) after considering the information provided to them (i.e., impact factors as seen in the previous task).

The results of the controlled group were as follows (after considering the impact information provided to them):

- Six students out of the eight who initially agreed to the estimate retained their original opinion by choosing "Equal to 8 story points". Two out of these eight students changed their opinion after considering the provided information (impact factors) and chose the estimate should be "Greater than 8 story points".
- One of the students who did not agree to the 8 story estimate changed his opinion and chose "Less than 8 story points".

To summarize, only two out of eight students after re-assessment of estimate changed their opinion whereas six remained indifferent. Perhaps in their view, the consideration of information does not influence the estimation.

On the other hand, when students of the treatment group were asked to re-assess the estimate of a feature after considering the information provided to them (i.e., impact factors as seen in the previous task). The results of the treatment group were as follows (after considering the impact information provided to them):

- Only one student out of the eight who initially agreed to the estimate, retained his/her original opinion by choosing "Equal to 8 story points". Five out of these eight students changed their opinion after considering the provided information (impact factors) and chose the estimate should be "Greater than

8 story points". One student chose the estimate should be "Significantly greater than 8 story points".

- One of the students chose the estimate should be "Significantly less than 8 story points".

To summarize, seven out of eight students after re-assessment of estimate changed their opinion whereas only one remained indifferent.

Students in the treatment group, after considering the additional information, seemed convinced that the information influences the estimates which were further confirmed in their rationale in Figure 5.10. By analyzing the open text answers from subjects, nine reasons were identified that they used to justify their effort-estimate (see Figure 5.10). One pattern is that subjects using HyEEASe have stated a broader variety of rationales and also have relied more on objective data to motivate their answers. This suggests that the subjects using HyEEASe were considering relevant data, as intended by the method when estimating the effort required to implement a feature. The strength of argumentation is seen as an indicator for a better understanding of a concept [20, 80].

The results indicate that the alternative hypothesis $H4_1$ could be accepted.

	A) Do you agree with the provided estimate for feature 1 (i.e. 8 SP)?		B) After considering the additional information, I would expect the estimate for feature 1 to be:				
			1. Significantly greater than 8 SP	2. Greater than 8 SP	3. Equal to 8 SP	4. Less than 8 SP	5. Significantly less than 8 SP
	Number of students		Number of students				
Grp A – Planning poker	Yes	8			6		
	No	1				1	
	I don't know	4		3	1		
Grp B – HyEEASe	Yes	8	1	5	1		
	No	1				1	
	I don't know	2					1

Figure 5.9: Testing the $H4$ - Learnability of estimation methods across both the groups

4. Perception of $H3$ (usefulness) and $H4$ (learnability): The perception of students on usefulness (understandability, completeness) and learnability of both the estimation methods were also captured using TAM concepts through a feedback questionnaire (Task T4 in Table 5.12) at the end of the experiment.

To quantify these aspects, a five-point Likert scale was used (scale: five-point ratio scale, with 5 being strongly agree and 1 being strongly disagree). The students were asked to rate the quality aspects on this scale after they have performed the tasks according

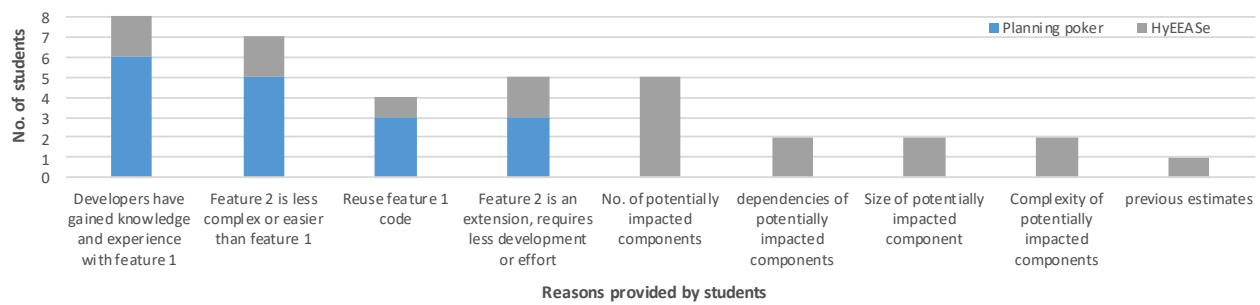


Figure 5.10: Students in both the groups providing rationale to justify their effort estimate

to the described estimation methods. Quality aspects rated with more than 3 (neutral opinion) show a positive evaluation overall.

Figure 5.11 shows the results of both the groups using boxplots. In the HyEEASe group, the spread is much closer which indicates a more positive perception about usefulness and learnability.

From the histograms in Figure 5.12, the across and within group analysis of both the estimation methods in their groups for usefulness and learnability can be observed. The histograms show that students in the HyEEASe group have rated the understandability, completeness, and learnability more positively as compared to the Planning Poker group. This indicates that the alternative hypotheses $H3_1$ and $H4_1$ could be accepted.

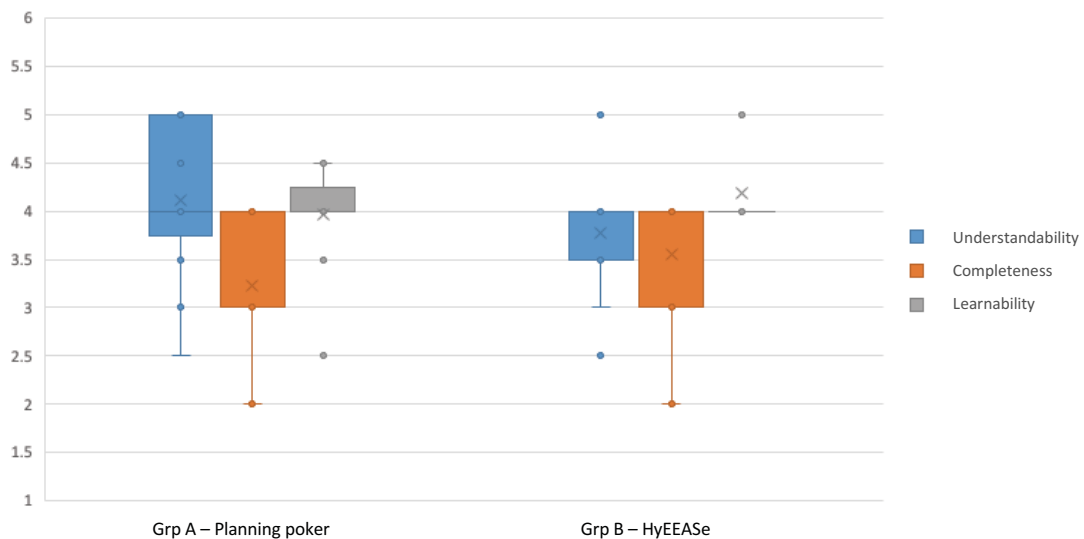


Figure 5.11: Perception of both groups on H3 (understandability, completeness) and H4 - learnability

Perception about H3 (understandability, completeness) and H4 (learnability) - Across and with in group analysis

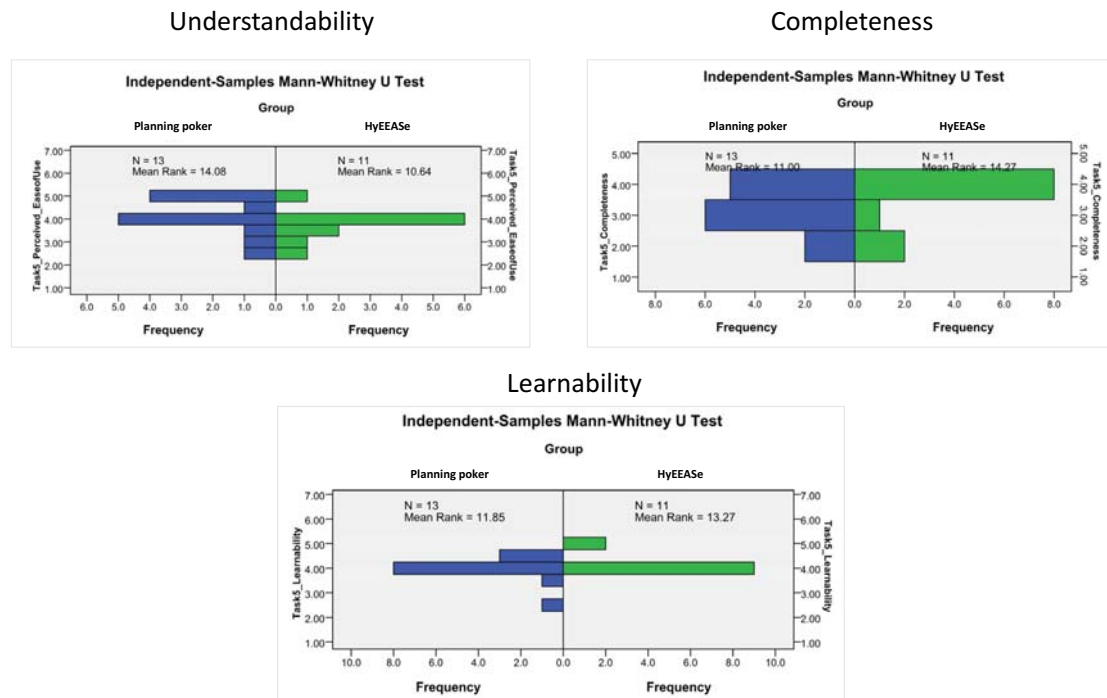


Figure 5.12: Perception about H3 (understandability, completeness) and H4 (learnability) across and with in the groups

The results of the subjective assessment of HyEEASe and Planning Poker by the students on “perceived ease of use”, “usefulness”, and “learnability” indicate positive results (see Figure 5.13). Relatively, Planning Poker was rated superior to HyEEASe on all dimensions. However, the positive rating of HyEEASe (see Figure 5.13) and the improved quality of argumentation for justifying the estimates (see Figure 5.10) indicates positive results for HyEEASe.

5.6.6 Conclusions

A controlled experiment was conducted with students taking the SPPM course at TUKL to determine the differences between two estimation methods, i.e., Planning Poker versus HyEEASe (Hybrid estimation) for H3: usefulness and H4: learnability hypotheses.

Stratified random sampling was used to divide students into two groups. The design was thus one factor (estimation method) with two treatments (i.e., Planning Poker versus HyEEASe). Each group was assigned the same set of tasks that were to be performed using their assigned estimation method. The evaluation of the tasks determined the useful-

		Grp B – HyEEASe						Grp A – Planning poker					
Perceived ease of use	I think the estimation method...	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	I do not know	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	I do not know
	1... is clear	0	0	4	6	1	0	0	0	2	6	5	0
	2... is understandable	0	0	2	7	2	0	0	0	1	4	8	0
	3... requires a lot of mental effort to apply it	0	5	3	1	2	0	0	2	3	4	4	0
	4... is easy to use	0	3	1	6	1	0	0	2	1	5	5	0
Usefulness	I think the information provided by change impact analysis is...												
	... informative	0	0	2	5	4	0	NA					
	... valuable	0	0	2	4	5	0						
	in general, useful for estimating effort for enhanced features	0	0	1	5	5	0						
Learnability	After using the estimation method ...												
	I learnt about the factors influencing estimation of enhanced features.	0	0	0	9	2	0	NA					
	I am convinced I can <i>learn over time</i> , which factors influence estimation of enhanced features.	0	0	0	7	4	0						
	I am convinced it will <i>help me learn fast about</i> the factors influencing estimation of enhanced features.	0	0	3	5	3	0						

Figure 5.13: Student feedback regarding understandability of the method

ness and learnability of both the estimation methods. The boxplots and histograms indicated that HyEEASe:

- Provides more understandable (impact) information than Planning Poker.
 - Indicated by no of the right answers while understanding the method.
 - Indicated by the positive perception in feedback questionnaire.
- Provides more complete (impact) information in comparison to Planning Poker.
 - Indicated by identifying information not yet considered by the method.
 - Indicated by the positive perception in feedback questionnaire.
- Enables learning of impact factors in comparison to Planning Poker.
 - Indicated by the change of opinion about an initial estimated effort.
 - Indicated by the rationale that led to their change of opinion.

- Indicated by the positive perception in feedback questionnaire.

After the experiment students on May 3rd, 2018 were presented with the design of the experiment and the execution process, as their right to service. All students were shown the experiment objective, format, both the estimation methods with workflow and the execution set up.

5.7 Summary

Table 5.15 shows the summary of all the above-mentioned evaluations conducted to evaluate HyEEASe with mock-up (only Hybrid estimation), HyEEASe (both Hybrid estimation and GBT model estimation) with respect to H1, H2, H3, and H4.

Table 5.15: Evaluation summary

Evaluation strategy	Evaluation object	H1 - Accuracy	H2 - Bias	H3 - Usefulness	H2 - Learnability
Case study at SAP SE	HyEEASe through mockup (only Hybrid estimation)	NA	NA	Positive perception on usefulness (understandable impact information)	NA
Case study at Insiders Technologies	HyEEASe (both Hybrid estimation and GBT model estimation)	Higher accuracy in comparison to Planning Poker	Lower bias in comparison to Planning Poker	Neutral perception on usefulness (understandable, useful, but not complete impact information)	Neutral perception on learnability (awareness)
Data-based evaluation	HyEEASe (only GBT model)	Notably higher accuracy in comparison to Agile COCOMO II	Notably lower bias in comparison to Agile COCOMO II	NA	NA
Controlled experiment with students	HyEEASe (only Hybrid estimation)	NA	NA	Provides more understandable and complete (impact) information in comparison to Planning Poker	Enables learning of impact factors in comparison to Planning Poker

The results of evaluating the HyEEASe Hybrid estimation through mock-up with SAP SE showed that the practitioners perceived HyEEASe useful for supporting EE. They also found the mock-up easy to use and intended to use it in their estimation process.

Encouraged by these results, the method was updated, in close collaboration with Insiders Technologies and a prototype was developed. Furthermore, the GBT model was also developed.

The results of evaluating the hybrid effort estimation method supported by a prototype with Insiders Technologies concluded that the practitioners also perceived the overall method and prototype as very useful for supporting EE. The impact information and the visualizations not only supported them during estimation but also enabled learning about the system. Furthermore, in their view, considering modified LOC and testing metrics among the impact factors could improve the effectiveness of IA for EE. Regarding estimation accuracy, the GBT model did not perform better than Hybrid estimation as the later provided additional useful impact information to the experts.

The data-based evaluation, in which the GBT model was compared with Agile COCOMO II concluded that the estimates produced by the GBT model have much higher accuracy and much lower bias than Agile COCOMO II.

The findings of the experiment conducted with students at TUKL indicated that HyEEASe provided more understandable and complete (impact) information as well as enabled learning of impact factors in comparison to Planning Poker.

Based on the evaluation results, it is concluded that both students and the practitioners perceived the overall method and the prototype as very useful for supporting EE. The impact information and the visualizations not only support them during estimation but also enable learning about the system.

It is, therefore, expected that the practitioners would be able to plan software releases better by making more informed decisions with more transparency during the estimation process. Revisiting the evaluations of estimation methods in agile software development in Table 5.16, we have found that HyEEASe outperformed the other proposed methods so far.

Table 5.16: Estimation methods evaluation

Estimation method	C1. Reliability	C2. Complexity	C3. Informative power	C4. Learning support	C5. Tool support	C6 Empirical evidence	C7. Data requirement	Reference
Planning poker	--	--	--	--	NA	-	-	[55][104][98][71][72]
Expert judgment	--	--	--	--	NA	-	-	[17][75][89][97][55][44]
Use Case Points (UCP) Method, UCP Method Modification	+	-	--	--	--	+	++	[89]
Linear regression, Neural Nets (SVM)	+	++	--	--	--	+	+	[17]
Robust regression, Neural Nets (RBF)	+	++	--	--	--	+	+	[16]
CAEA	-	+	--	--	--	--	++	[32]
SVR: Linear Kernel, Polynomial Kernel, RBF Kernel, Sigmoid Kernel	-	++	--	--	--	-	++	[101]
GRNN, PNN, GMDH, Cascade-Correlation Neural Network	-	++	--	--	--	+	++	[100]
Kalman filter	-	++	--	--	--	+	+	[65]
Cosmic FP	-	+	--	--	--	+	-	[58]
Statistical combination of individual estimates	+	-	--	--	--	+	-	[76]
Own algorithm	-	-	--	--	--	--	++	[92]
HyEEASe	++	--	++	++	++	++	--	[109]

6 Conclusions and future work

This chapter revisits the contributions and discusses future work for this thesis. This thesis introduced HyEEASe - a hybrid, lightweight and systematic method for estimating the effort in the context of iterative, incremental software development. The existing methods have limitations in terms of no objective consideration of the potential impact of a change on existing software and custom factors that contribute to the effort overhead. This thesis, therefore proposed a method to address these issues and conducted a set of studies to evaluate it. The main scientific and empirical contributions of this thesis are as follows:

1. An industrial case study and a survey using interviews and questionnaires were conducted for establishing the state-of-the-practice of estimation methods at the case companies, i.e., SAP SE and Insiders Technologies GmbH respectively. Through these studies, the needs of industry practitioners, and their challenges and development contexts were understood. The industrial objectives and requirements were taken as input from these studies, and the results of systematic literature reviews were used in the development of the method.
2. Literature reviews were conducted for establishing the state-of-the-art for change impact analysis and effort estimation in agile development. As a result of these reviews, it was recognized that expert-based estimation methods are the most dominant methods used in an agile development context. Moreover, these methods are unreliable, utilize limited information and do not support systematic learning. It was also recognized that the research work on estimation methods is not driven by industrial objectives. Existing estimation methods have focused on improving the accuracy of estimates only, while not providing a systematic analysis of useful information eventually supporting practitioners in making informed and unbiased decisions. In other words, these estimation methods do not objectively consider the potential impact of a change on existing software that contributes to the effort overhead. As a result, practitioners remain unaware of the factors that influence estimation, and inaccurate estimates are produced that further lead to schedule and budget overruns. Experts remain unaware of the factors that influence estimation.
3. This led to the development of HyEEASe. HyEEASe is a hybrid, lightweight and systematic method for estimating the effort in the context of iterative and incremental software development. The

method is a result of synthesizing research on effort estimation and change impact analysis. Static code analysis and historical change impact analysis were used to isolate the potentially impacted parts for a change. Furthermore, past estimation and data related to the impacted parts were also presented to practitioners while they estimate effort for a new change. The key elements of the method included: 1) Analysis of affected software concerning the impact of the change. 2) Measurement of size and complexity of change. 3) Providing historical estimates and impact information for similar impacted parts. 4) Visualizing the impact information for the experts to interact with. 5) Providing an experience base to store impact and estimation data for organizational learning and future reuse. Additionally, a Gradient Boosted Trees (GBT) based estimation model was also developed.

4. The development of a prototype tool to support the hybrid method as well as the GBT estimation model.
5. Two case studies at two companies were conducted. The first case study was performed at SAP SE, a German multinational software corporation, to evaluate the perceived usefulness and effectiveness of the concept through mock-ups. The results indicated HyEEASe useful for supporting effort estimation. The mock-up was easy to use and provided useful visualization capabilities and practitioners expressed interest to use it in their estimation process. The second case study was done at Insiders Technologies GmbH, where the effectiveness, the perceived usefulness, and the learnability of the refined hybrid method through using the prototype tool was evaluated. The results of this case study showed that the proposed method produced more accurate estimates than purely expert-based or purely model-based estimates.
6. A controlled experiment was conducted with the BS and MS students taking Software Process and Project Management (SPPM) course at the Department of Computer Science, Technische Universität Kaiserslautern (TUKL) to find the usefulness and learnability of the proposed hybrid method. The results of the experiment also indicated that the hybrid approach provides more useful impact information and enables learning of impact information as compared to pure expert-based estimation approaches like planning poker.

Like other research works, this research also has certain limitations as mentioned in Section 4.9. For example, HyEEASe currently only considers a limited number of OO code metrics like impact factors, i.e., size in LOC/SLOC, cyclomatic complexity, and coupling between objects, yet the method is not restricted to these metrics only.

Similarly, the used impact analysis technique was an adapted version of a classic combination of structural static and historical analysis. This

might not be the best technique in terms of the accuracy of the impact set. The feasibility of the approach as demonstrated in the thesis opens future directions for research where we can explore the use of more advanced changed impact analysis techniques. Dynamic impact analysis techniques though are expensive yet are more accurate in computing the impact set. In future studies, visualization of the impact information can also be utilized to better communicate the impact information for a change. Several software visualization techniques for rendering the structure and behavior of a software system have been proposed.

Moreover, we would also like to conduct a longitudinal study to verify one of the hypotheses that we could not evaluate during this research, i.e., Hypothesis H5: HyEEASe enables feedback across multiple iterations to learn about discrepancies between estimated and actual effort. We will need to use the method for multiple sprints in the company to find the learnability of the experts regarding impact factors and their effect on estimated effort.

Though various evaluations were carried out to validate different hypotheses of the method, yet convenience sampling and a small sample size reduce the statistical generalizability of our results. However, given the consistent positive evaluation from both students and practitioners participating in the studies gives confidence that the method is understandable. Nonetheless, to increase the representativeness of our results, we will need to use the method for multiple sprints in the company. More replications in different companies will improve the generalizability of the results. The general instance of HyEEASe, however, will need to be operationalized in each case company. However, the main limitation of the applicability of the method is that it will not apply to completely new development projects.

This thesis has utilized two complementary research areas to address a practical problem for software companies. By leveraging existing information in software repositories available in most companies and integrating it effectively with the most often used effort estimation method, the proposal put forth in this thesis has large potential for impact.

In modern software development where the focus is on shorter iteration and delivering value faster to the customer. The ability to structure development in these short iterations is paramount to this paradigm. All long-term, strategic planning, and road mapping relies eventually on the ability to deliver at the sprint level.

The impact of this thesis is not just to the point estimates produced as an outcome of the estimation activity. Rather the impact information generated and effectively represented to the practitioners provides the possibility to work on the scoping decision of a sprint ("what should go in a sprint"), is the requirement, feature or user story too big, should it be broken down further, have we involved the right resources who should be estimating and also working on the requirement. Thus, it has the

potential to not only improve the point estimates but more importantly improve an organization's ability to plan and organize the operational aspects of software development.

Furthermore, instead of delving further in the discussion, whether a model-based or expert-based models for estimation are the way forward, the proposed combination of change impact analysis with effort estimation takes an effective step to improve the accuracy of Planning Poker. This also provides a practical use case for mining software repository research that is sometimes criticized for being opportunistic and targeting the improvement of practice.

7 References

- [1] Center of System and Software Engineering - Agile Methods Agile COCOMO® II. URL <http://csse.usc.edu/csse/research/AgileCOCOMO/>. Accessed: 2018-03-01.
- [2] Balsamiq Mockups. URL <https://balsamiq.com/products/mockups/>. Accessed: 2017-03-14.
- [3] Insiders Technologies company. URL <https://www.insiders-technologies.de/en/home/company.html>. Accessed: 2018-10-14.
- [4] Gradient Boosted Trees. URL https://docs.rapidminer.com/latest/studio/operators/modeling/predictive/trees/gradient_boosted_trees.html. Accessed: 2018-01-14.
- [5] Planning poker, Accessed: 2016-03-14. URL <https://www.mountaingoatsoftware.com/agile/planning-poker>.
- [6] MAXQDA - Qualitative Data Analysis Software, Accessed: 2017-03-14. URL <http://www.maxqda.com/>.
- [7] SAP About SAP company, Accessed: 2017-03-14. URL <https://www.sap.com/corporate/en/company.html>.
- [8] COCOMO II Model Definition Manual, Accessed: 2018-03-01. URL http://sunset.usc.edu/csse/affiliate/private/COCOMOII_2000_3/modelman.pdf.
- [9] Estimation game, Accessed: 2018-03-01. URL <http://www.agilelearninglabs.com/2012/05/how-to-play-the-team-estimation-game/>.
- [10] IBM SPSS, Accessed: 2018-03-01. URL <http://www-01.ibm.com/software/de/analytics/spss/products/statistics/>.
- [11] JIRA tool, Accessed: 2018-03-01. URL <https://www.atlassian.com/software/jira>.
- [12] Understand tool, Accessed: 2018-03-01. URL <https://scitools.com/features/>.
- [13] Moodle, Accessed: 2018-03-14. URL <https://moodle.org/>.
- [14] Google Scholar, Accessed: 2019-02-14. URL <https://scholar.google.com/>.
- [15] Scopus, Accessed: 2019-02-14. URL <https://www.scopus.com/>.
- [16] P. Abrahamsson, R. Moser, W. Pedrycz, A. Sillitti, and G. Succi. Effort prediction in iterative software development processes - incremental versus global prediction models. In Proceedings of the First International Symposium on Empirical Software Engineering and Measurement, ESEM, pages 344–353, 2007.

- [17] P. Abrahamsson, I. Fronza, R. Moser, J. Vlasenko, and W. Pedrycz. Predicting development effort from user stories. In *Proceedings of the International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 400–403. IEEE, 2011.
- [18] N. B. Ali. Is effectiveness sufficient to choose an intervention?: Considering resource use in empirical software engineering. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM*, pages 54:1–54:6, 2016.
- [19] N. B. Ali and K. Petersen. Evaluating strategies for study selection in systematic literature studies. In *Proceedings of the International Symposium on Empirical Software Engineering and Measurement, ESEM*, pages 45:1–45:4, 2014.
- [20] N. B. Ali and M. Unterkalmsteiner. Use and evaluation of simulation for software process education: a case study. In *Proceedings of the European Conference Software Engineering Education (ECSEE)*, pages 59–73. Shaker Verlag, 2014.
- [21] N. B. Ali and M. Usman. Reliability of search in systematic reviews: Towards a quality assessment framework for the automated-search strategy. *Information & Software Technology*, 99:133–147, 2018.
- [22] N. B. Ali, K. Petersen, and M. Mäntylä. Testing highly complex system of systems: an industrial case study. In *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM '12*, pages 211–220, 2012.
- [23] N. B. Ali, K. Petersen, and K. Schneider. Flow-assisted value stream mapping in the early phases of large-scale software development. *Journal of Systems and Software*, 111:213–227, 2016.
- [24] N. B. Ali, E. Engström, M. Taromirad, M. R. Mousavi, N. M. Minhas, D. Helgesson, S. Kunze, and M. Varshosaz. On the search for industry-relevant regression testing research. *Empirical Software Engineering*, 24(4):2020–2055, 2019.
- [25] G. Antoniol, G. Canfora, and A. De Lucia. Estimating the size of changes for evolving object oriented systems: A case study. In *Proceedings of the 6th International Symposium on Software Metrics*, pages 250–258. IEEE, 1999.
- [26] T. Apiwattanapong, A. Orso, and M. J. Harrold. Efficient and precise dynamic impact analysis using execute-after sequences. In *Proceedings of the 27th International Conference on Software Engineering (ICSE)*, pages 432–441, 2005.
- [27] K. Auer and R. Miller. *Extreme programming applied: playing to win*. Addison-Wesley Longman Publishing Co., Inc., 2001.
- [28] L. Badri, M. Badri, and D. St-Yves. Supporting predictive change impact analysis: A control call graph based technique. In *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC)*, pages 167–175. IEEE, 2005.
- [29] V. R. Basili. The experience factory and its relationship to other quality approaches. *Advances in Computers*, 41:65–82, 1995.

- [30] K. Beck, M. Beedle, A. Van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, et al. Agile manifesto, 2001. URL <http://www.agilemanifesto.org>, 2009.
- [31] Á. Beszédes, T. Gergely, S. Farago, T. Gyimóthy, and F. Fischer. The dynamic function coupling metric and its use in software evolution. In *Proceedings of the 11th European Conference on Software Maintenance and Reengineering, Software Evolution in Complex Software Intensive Systems, CSMR*, pages 103–112, 2007.
- [32] S. Bhalerao and M. Ingle. Incorporating vital factors in agile estimation through algorithmic method. *International Journal of Computer Science & Applications*, 6(1):85–97, 2009.
- [33] B. Boehm, B. Clark, E. Horowitz, C. Westland, R. Madachy, and R. Selby. Cost models for future software life cycle processes: Cocomo 2.0. *Annals of software engineering*, 1(1):57–94, 1995.
- [34] B. Boehm, C. Abts, B. Clark, and S. Devnani-Chulani. Cocomo ii model definition manual. Technical report, The University of Southern California, 1997.
- [35] S. A. Bohner. Impact analysis in the software change process: a year 2000 perspective. In *Proceedings of the International Conference on Software Maintenance (ICSM)*, pages 42–51, 1996.
- [36] B. Breech, A. Danalis, S. A. Shindo, and L. L. Pollock. Online impact analysis via dynamic compilation technology. In *Proceedings of the 20th International Conference on Software Maintenance (ICSM)*, pages 453–457, 2004.
- [37] B. Breech, M. Tegtmeyer, and L. L. Pollock. A comparison of online and dynamic impact analysis algorithms. In *Proceedings of the 9th European Conference on Software Maintenance and Reengineering (CSMR)*, pages 143–152, 2005.
- [38] L. C. Briand, K. E. Emam, and F. Bomarius. COBRA: A hybrid method for software cost estimation, benchmarking, and risk assessment. In *Proceedings of the 1998 International Conference on Software Engineering, ICSE 98*, Kyoto, Japan, April 19-25, 1998., pages 390–399, 1998.
- [39] L. C. Briand, J. Wust, and H. Lounis. Using coupling measurement for impact analysis in object-oriented systems. In *Proceedings of the IEEE International Conference on Software Maintenance, (ICSM)*, pages 475–482. IEEE, 1999.
- [40] L. C. Briand, J. Wüst, and H. Lounis. Replicated case studies for investigating quality factors in object-oriented designs. *Empirical software engineering*, 6(1):11–58, 2001.
- [41] J. Buckner, J. Buchta, M. Petrenko, and V. Rajlich. Jripples: A tool for program comprehension during incremental change. In *Proceedings of the 13th International Workshop on Program Comprehension (IWPC)*, pages 149–152, 2005.

- [42] G. Canfora and L. Cerulo. Impact analysis by mining software and change request repositories. In *Proceedings of the 11th IEEE International Symposium on Software Metrics (METRICS)*, page 29, 2005.
- [43] L. Cao. Estimating agile software project effort: An empirical study. In *Learning from the past & charting the future of the discipline. Proceedings of the 14th Americas Conference on Information Systems, AMCIS*, page 401, 2008.
- [44] C. Catal and M. S. Aktas. A composite project effort estimation approach in an enterprise software development project. In *Proceedings of the 23rd International Conference on Software Engineering & Knowledge Engineering (SEKE)*, pages 331–334, 2011.
- [45] S. R. Chidamber and C. F. Kemerer. A metrics suite for object oriented design. *IEEE Transactions on software engineering*, 20(6):476–493, 1994.
- [46] B. Dit, M. Wagner, S. Wen, W. Wang, M. L. Vásquez, D. Poshyvanyk, and H. H. Kagdi. Impactminer: a tool for change impact analysis. In *Companion Proceedings of the 36th International Conference on Software Engineering, ICSE*, pages 540–543, 2014.
- [47] M. O. Elish, H. I. Aljamaan, and I. Ahmad. Three empirical studies on predicting software maintainability using ensemble methods. *Soft Comput.*, 19(9):2511–2524, 2015.
- [48] K. R. Felizardo, E. Mendes, M. Kalinowski, É. F. Souza, and N. L. Vijaykumar. Using forward snowballing to update systematic reviews in software engineering. In *Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, page 53. ACM, 2016.
- [49] T. Foss, E. Stensrud, B. A. Kitchenham, and I. Myrtveit. A simulation study of the model evaluation criterion MMRE. *IEEE Trans. Software Eng.*, 29(11):985–995, 2003.
- [50] M. Gethers, B. Dit, H. Kagdi, and D. Poshyvanyk. Integrated impact analysis for managing software changes. In *Proceedings of the 34th International Conference on Software Engineering (ICSE)*, pages 430–440. IEEE, 2012.
- [51] D. L. Goodhue and R. L. Thompson. Task-technology fit and individual performance. *MIS Quarterly*, pages 213–236, 1995.
- [52] S. Gwizdala, Y. Jiang, and V. Rajlich. Jtracker - A tool for change propagation in java. In *Proceedings of the 7th European Conference on Software Maintenance and Reengineering (CSMR)*, pages 223–229, 2003.
- [53] L. Hattori, G. dos Santos Jr, F. Cardoso, and M. Sampaio. Mining software repositories for software change impact analysis: a case study. In *Proceedings of the 23rd Brazilian symposium on Databases*, pages 210–223. Sociedade Brasileira de Computação, 2008.

- [54] L. Hattori, D. Guerrero, J. Figueiredo, J. Brunet, and J. Damásio. On the precision and accuracy of impact analysis techniques. In *Proceedings of the 7th IEEE/ACIS International Conference on Computer and Information Science, ICIS*, pages 513–518. IEEE, 2008.
- [55] N. C. Haugen. An empirical study of using planning poker for user story estimation. In *Proceedings of AGILE Conference (AGILE)*, pages 23–34, 2006.
- [56] P. Hearty, N. E. Fenton, D. Marquez, and M. Neil. Predicting project velocity in XP using a learning dynamic bayesian network model. *IEEE Trans. Software Eng.*, 35(1):124–137, 2009.
- [57] L. Huang and Y.-T. Song. Precise dynamic impact analysis with dependency analysis for object-oriented programs. In *Proceedings of the 5th ACIS International Conference on Software Engineering Research, Management & Applications (SERA)*, pages 374–384. IEEE, 2007.
- [58] I. Hussain, L. Kosseim, and O. Ormandjieva. Approximation of COSMIC functional size to support early effort estimation in agile. *Data Knowl. Eng.*, 85:2–14, 2013.
- [59] R. Jabbari, N. B. Ali, K. Petersen, and B. Tanveer. What is devops?: A systematic mapping study on definitions and practices. In *Proceedings of the Scientific Workshop Proceedings of XP2016, Edinburgh, Scotland, UK, May 24, 2016*, page 12, 2016.
- [60] R. Jabbari, N. B. Ali, K. Petersen, and B. Tanveer. Towards a benefits dependency network for devops based on a systematic literature review. *Journal of Software: Evolution and Process*, 30(11), 2018.
- [61] A. Jedlitschka and D. Pfahl. Reporting guidelines for controlled experiments in software engineering. In *Proceedings of the International Symposium on Empirical Software Engineering (ISESE)*, pages 95–104. IEEE, 2005.
- [62] M. Jørgensen. A review of studies on expert estimation of software development effort. *Journal of Systems and Software*, 70(1):37–60, 2004.
- [63] M. Jørgensen, B. W. Boehm, and S. Rifkin. Software development effort estimation: Formal models or expert judgment? *IEEE Software*, 26(2):14–19, 2009.
- [64] H. H. Kagdi, M. Gethers, D. Poshyvanyk, and M. L. Collard. Blending conceptual and evolutionary couplings to support change impact analysis in source code. In *Proceedings of the 17th Working Conference on Reverse Engineering, WCRE*, pages 119–128, 2010.
- [65] S. Kang, O. Choi, and J. Baik. Model-based dynamic cost estimation and tracking method for agile software development. In *Proceedings of the IEEE/ACIS 9th International Conference on Computer and Information Science (ICIS)*, pages 743–748. IEEE, 2010.
- [66] B. A. Kitchenham, L. M. Pickard, S. G. MacDonell, and M. J. Shepperd. What accuracy statistics really measure. *IEE Proceedings-Software*, 148(3):81–85, 2001.

- [67] J. Law and G. Rothermel. Whole program path-based dynamic impact analysis. In *Proceedings of the 25th International Conference on Software Engineering*, pages 308–318, 2003.
- [68] Y. W. Lee and D. M. Strong. Knowing-why about data processes and data quality. *Journal of Management Information Systems*, 20(3):13–39, 2003.
- [69] S. Lehnert. A taxonomy for software change impact analysis. In *Proceedings of the 12th International Workshop on Principles of Software Evolution and the 7th annual ERCIM Workshop on Software Evolution*, pages 41–50. ACM, 2011.
- [70] V. Lenarduzzi. Could social factors influence the effort software estimation? In *Proceedings of the 7th International Workshop on Social Software Engineering, SSE*, pages 21–24, 2015.
- [71] V. Lenarduzzi and D. Taibi. Can functional size measures improve effort estimation in scrum. In *Proceedings of the International Conference on Software Engineering and Advances, ICSEA*, 2014.
- [72] V. Lenarduzzi, I. Lunesu, M. Matta, and D. Taibi. Functional size measures and effort estimation in agile development: A replicated study. In *Proceedings of the 16th International Conference on Agile Processes in Software Engineering and Extreme Programming, XP*, pages 105–116, 2015.
- [73] B. Li, X. Sun, H. Leung, and S. Zhang. A survey of code-based change impact analysis techniques. *Softw. Test., Verif. Reliab.*, 23(8):613–646, 2013.
- [74] W. Li and S. Henry. Object-oriented metrics that predict maintainability. *Journal of systems and software*, 23(2):111–122, 1993.
- [75] K. Logue, K. McDaid, and D. Greer. Allowing for task uncertainties and dependencies in agile release planning. In *Proceedings of the Software Measurement European Forum (SMEF)*, 2007.
- [76] V. Mahnic and T. Hovelja. On using planning poker for estimating user stories. *Journal of Systems and Software*, 85(9):2086–2095, 2012.
- [77] R. C. Martin. *Clean code: a handbook of agile software craftsmanship*. Pearson Education, 2009.
- [78] V. R. McKinney, K. Yoon, and F. Zahedi. The measurement of web-customer satisfaction: An expectation and disconfirmation approach. *Information Systems Research*, 13(3):296–315, 2002.
- [79] N. M. Minhas, K. Petersen, N. B. Ali, and K. Wnuk. Regression testing goals - view of practitioners and researchers. In *24th Asia-Pacific Software Engineering Conference Workshops, APSEC Workshops 2017, Nanjing, China, December 4-8, 2017*, pages 25–31, 2017.
- [80] J. S. Molléri, N. B. Ali, K. Petersen, N. M. Minhas, and P. Chatzipetrou. Teaching students critical appraisal of scientific literature using checklists. In *Proceedings of the 3rd European Conference of Software Engineering Education, ECSEE*, pages 8–17, 2018.

- [81] K. Moløkken-Østvold and K. M. Furulund. The relationship between customer collaboration and software project overruns. In *Proceedings of the AGILE 2007 Conference (AGILE)*, pages 72–83, 2007.
- [82] K. Moløkken-Østvold and M. Jørgensen. A comparison of software project overruns-flexible versus sequential development models. *IEEE Transactions of Software Engineering*, 31(9):754–766, 2005.
- [83] K. Moløkken-Østvold, M. Jørgensen, S. S. Tanilkan, H. Gallis, A. C. Lien, and S. E. Hove. A survey on software estimation in the norwegian industry. In *Proceedings of the 10th IEEE International Software Metrics Symposium (METRICS)*, pages 208–219, 2004.
- [84] R. Moser, W. Pedrycz, and G. Succi. Incremental effort prediction models in agile development using radial basis functions. In *Proceedings of the 19th International Conference on Software Engineering & Knowledge Engineering (SEKE)*, pages 519–522, 2007.
- [85] A. B. Nassif, L. F. Capretz, D. Ho, and M. Azzeh. A treeboost model for software effort estimation based on use case points. In *Proceedings of the 11th International Conference on Machine Learning and Applications, ICMLA*, pages 314–319. IEEE, 2012.
- [86] R. R. Nelson, P. A. Todd, and B. H. Wixom. Antecedents of information and system quality: An empirical examination within the context of data warehousing. *J. of Management Information Systems*, 21(4):199–236, 2005.
- [87] S. P. Nerur, R. Mahapatra, and G. Mangalaraj. Challenges of migrating to agile methodologies. *Commun. ACM*, 48(5):72–78, 2005.
- [88] D. Nguyen-Cong and D. Tran-Cao. A review of effort estimation studies in agile, iterative and incremental software development. In *Proceedings of the IEEE International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future, RIVF*, pages 27–30, 2013.
- [89] A. W. M. M. Parvez. Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development. In *Proceedings of the International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 113–118. IEEE, 2013.
- [90] K. Petersen and N. B. Ali. Identifying strategies for study selection in systematic reviews and maps. In *Proceedings of the 5th International Symposium on Empirical Software Engineering and Measurement, ESEM*, pages 351–354, 2011.
- [91] M. Petrenko and V. Rajlich. Variable granularity for improving precision of impact analysis. In *Proceedings of the 17th International Conference on Program Comprehension*, pages 10–19. IEEE, 2009.
- [92] R. Popli and N. Chauhan. Cost and effort estimation in agile software development. In *Proceedings of the International Conference on Optimization, Reliability, and Information Technology (ICROIT)*, pages 57–61. IEEE, 2014.

- [93] D. Poshyvanyk, A. Marcus, R. Ferenc, and T. Gyimóthy. Using information retrieval based coupling measures for impact analysis. *Empirical Software Engineering*, 14(1):5–32, 2009.
- [94] A. T. Raslan, N. R. Darwish, and H. A. Hefny. Towards a fuzzy based framework for effort estimation in agile software development. *International Journal of Computer Science and Information Security*, 13(1):37, 2015.
- [95] X. Ren, F. Shah, F. Tip, B. G. Ryder, and O. C. Chesley. Chianti: a tool for change impact analysis of java programs. In *Proceedings of the 19th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA*, pages 432–448, 2004.
- [96] P. Runeson and M. Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, 2009.
- [97] D. Sande, A. Sanchez, R. Montebelo, S. Fabbri, and E. M. Hernandez. Pw-plan - A strategy to support iteration-based software planning. In *Proceedings of the 12th International Conference on Enterprise Information Systems, ICEIS*, pages 66–74, 2010.
- [98] C. Santana, F. Leoneo, A. Vasconcelos, and C. Gusmão. Using function points in agile projects. In *Proceedings of the 12th International Conference on Agile Processes in Software Engineering and Extreme Programming, XP*, pages 176–191, 2011.
- [99] F. M. Santos and K. M. Eisenhardt. Multiple case study. *Encyclopedia of Social Science Research Methods*, pages 685–686, 2004.
- [100] S. M. Satapathy and S. K. Rath. Empirical assessment of machine learning models for agile software development effort estimation using story points. *Innovations in Systems and Software Engineering*, 13(2-3):191–200, 2017.
- [101] S. M. Satapathy, B. P. Acharya, and S. K. Rath. Class point approach for software effort estimation using stochastic gradient boosting technique. *ACM SIGSOFT Software Engineering Notes*, 39(3):1–6, 2014.
- [102] S. M. Satapathy, A. Panda, and S. K. Rath. Story point approach based agile software effort estimation using various SVR kernel methods. In *Proceedings of the 26th International Conference on Software Engineering and Knowledge Engineerings*, pages 304–307, 2014.
- [103] A. Sharma and D. S. Kushwaha. Applying requirement based complexity for the estimation of software development and testing effort. *ACM SIGSOFT Software Engineering Notes*, 37(1):1–11, 2012.
- [104] R. Tamrakar and M. Jørgensen. Does the use of fibonacci numbers in planning poker affect effort estimates? In *Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering, EASE*, pages 228–232, 2012.

- [105] B. Tanveer. Guidelines for utilizing change impact analysis when estimating effort in agile software development. In *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering, EASE*, 2017.
- [106] B. Tanveer, L. Guzmán, and U. M. Engel. Understanding and improving effort estimation in agile software development: An industrial case study. In *Proceedings of the International Conference on Software and Systems Process, ICSSP '16*, pages 41–50, 2016. ISBN 978-1-4503-4188-2.
- [107] B. Tanveer, L. Guzmán, and U. M. Engel. Effort estimation in agile software development: Case study and improvement framework. *Journal of Software: Evolution and Process*, 29(11), 2017.
- [108] B. Tanveer, A. M. Vollmer, and U. M. Engel. Utilizing change impact analysis for effort estimation in agile development. In *Proceedings of the 43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA*, pages 430–434, 2017.
- [109] B. Tanveer, A. M. Vollmer, and S. Braun. A hybrid methodology for effort estimation in agile development: an industrial evaluation. In *Proceedings of the 2018 International Conference on Software and System Process*, pages 21–30. ACM, 2018.
- [110] M. Torchiano and F. Ricca. Impact analysis by means of unstructured knowledge in the context of bug repositories. In *Proceedings of the International Symposium on Empirical Software Engineering and Measurement, ESEM*, page 47. ACM, 2010.
- [111] A. Trendowicz. Software effort estimation with well-founded causal models. PhD thesis, Technische Universität Kaiserslautern, Germany, 2008.
- [112] A. Trendowicz and R. Jeffery. *Software Project Effort Estimation - Foundations and Best Practice Guidelines for Success*. Springer, 2014.
- [113] M. Usman, E. Mendes, F. Weidt, and R. Britto. Effort estimation in agile software development: a systematic literature review. In *Proceedings of the 10th International Conference on Predictive Models in Software Engineering, PROMISE*, pages 82–91, 2014.
- [114] M. Usman, E. Mendes, and J. Börstler. Effort estimation in agile software development: a survey on the state of the practice. In *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering, EASE 2015, Nanjing, China, April 27-29, 2015*, pages 12:1–12:10, 2015.
- [115] R. Van Solingen, V. Basili, G. Caldiera, and H. D. Rombach. Goal question metric (gqm) approach. *Encyclopedia of Software Engineering*, 2002.
- [116] V. Venkatesh and H. Bala. Technology acceptance model 3 and a research agenda on interventions. *Decision sciences*, 39(2):273–315, 2008.
- [117] C. Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE*, pages 38:1–38:10, 2014.

- [118] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. Experimentation in software engineering. Springer Science & Business Media, 2012.
- [119] R. Woolson. Wilcoxon signed-rank test. Wiley Encyclopedia of Clinical Trials, 2008.
- [120] S. K. T. Ziauddin and S. Zia. An effort estimation model for agile software development. Advances in computer science and its applications (ACSA), 314:314–324, 2012.
- [121] T. Zimmermann, A. Zeller, P. Weissgerber, and S. Diehl. Mining version histories to guide software changes. IEEE Transactions on Software Engineering, 31(6): 429–445, 2005.

A Appendix A

This appendix contains the informed consent, observational protocol and the questionnaire for establishing the state-of-the-practice at SAP SE. Some questions in the questionnaire were adapted from [111].

Case study at SAP SE
**Informed content for observations and interviews for establishing state-of-the-practice
of effort estimation at SAP SE**

Informed Consent Form for development teams at SAP who we are inviting to participate in the observation sessions to be held in few selected scrum estimating meetings and face to face or skype/tele interviews. The purpose of these observations and interviews is to establish the state-of-the-practice of effort estimation of change requests process at SAP.

This Informed Consent Form has two parts:

- Information Sheet (to share information about the study with you)
- Certificate of Consent (for signatures if you choose to participate)

You will be given a copy of the full Informed Consent Form

Part I: Information Sheet

Introduction

I am Binish Tanveer, working in the Process, Compliance and Improvement department at Fraunhofer IESE. I am doing research on improving the process of effort estimation in Agile software development methodologies. In particular, I am currently characterizing the effort estimation of change requests in the context of sprints in Scrum. A part of this research will be conducted in the context of Software Campus project initiative, funded by BMBF number 01IS12053 titled “HyEEASe – Hybrid Effort Estimation of increments in Agile and Incremental Software Development”. I am going to give you information and invite you to be part of this research.

If you have questions, you can contact me.

Purpose of the research

In agile development, the requirements typically cannot be completely specified upfront but are developed as the project progresses and therefore need to be adjusted for in every increment (or release). In such an environment, systematic cost and effort estimation is challenging. Currently, effort estimation in this context relies heavily on human judgment. A cross-functional team of experts estimate by consensus how much effort a certain change will entail. Expert-based estimation not only costs a significant amount of effort but also has limited prediction accuracy due to the use of limited information (subjective judgment) and human judgment bias (individual and group effects). Moreover, in agile development this approach does not consider the potential impact of a change in software artifacts, which makes effort estimates obtained by using such methods less reliable. Therefore, high improvement potential exists with respect to systematic effort estimation in this environment.

Consequently, the purpose of my current research is to:

Characterize the effort estimation process of change requests
with respect to the effectiveness
in the context of agile software development
from the perspective of SAP development teams.

We believe that you can help us to better understand how agile development teams estimate effort of implementing change requests. In particular, we want to learn:

1. What information and artefacts are used while making estimates? For example, historical data, documents, factors (e.g. developer experience...), test cases, code.
2. What is the decision criterion for making estimates?
 - for selecting change requests (prioritization)
 - for estimating change requests (effort)
3. What are the rationales behind the decision criterion?
4. What are the outcomes of the estimation process?
5. How the outcomes of the estimation process are documented and communicated?
6. What are the strengths and weaknesses of prevailing estimation methods as well as the needs for improving them?

Research Intervention

This research takes the form of an observational and interview study in which two researchers will observe a series of effort estimation meetings. The observations will be conducted as part of your Scrum sessions. Interviews will be held either face to face or skype/ telephone based on the availability of the participant.

The study includes the following activities:

Before the study

- An informal consent describing the goals of this study will be sent over an email to SAP administration for SAP development teams.

During the study

- Two researchers will take notes collecting information about the effort estimation process including:
 - Roles participating in effort estimation process
 - Input to estimation process
 - Intermediate steps taken
 - Decision made with certain criteria and rationale
 - Outcomes of the estimation process

After the study

- After each observation session or an interview
 - Five-ten minutes question and answer round for clarification of queries.
- Feedback session (at the end of whole observational and interview study)
 - Presentation of study results.
 - Participant required: at least one representative of each team (all will be invited though)

The scheduling of the above-mentioned study will be coordinated according to the availability of the development teams.

Participant Selection

You are being invited to take part in this research because we feel that your experience as agile developer, tester or product owner can contribute much to our understanding of state-of-the-practice of effort estimation in agile at SAP. Your participation in this research is entirely voluntary.

Confidentiality

The information (i.e. raw data that is collected directly from observations or interviews) that we collect from this observational study will be kept confidential. It will not be shared with or given to anyone except the research team introduced at the beginning of the present document. Any information about you will have a number on it instead of your name. Only the researchers will know what your number is and we will lock that information up with a lock and key.

Sharing the Results

1. **Information** (i.e. raw data that is collected directly from observations/ interviews)
Nothing that you tell us will be shared with anybody outside the research team even inside SAP and the participants of this research, and nothing will be attributed to you by name.
2. **Knowledge** (i.e. the aggregated results of observational sessions and interviews)
The knowledge that we get from this study will be shared with you and the other participants before it is made widely available to the public. Each participant will receive a summary of the results. This knowledge study will be published as part of Binish Tanveer's PhD Thesis. In addition, we will publish the results in order that other interested people may learn from our research e.g. conferences, journals.

Right to refuse or withdraw

Participation in this study is entirely voluntary. It is your choice whether to participate or not. You do not have to take part in this research if you do not wish to do so, and choosing to participate will not affect your job or job-related evaluations in any way. You may stop participating in the study at any time that you wish without your job being affected.

Who to Contact

If you have any questions you may ask them now or later, even after the study has started. If you wish to ask questions later, you may contact: Binish Tanveer (binish.tanveer@iese.fraunhofer.de)

Part II: Certificate of Consent

I have been invited to participate in the observational and interview study about "Effort estimation of changes in Agile software development at SAP".

I have read the foregoing information, or it has been read to me. I have had the opportunity to ask questions about it and any questions I have been asked and have been answered to my satisfaction. I consent voluntarily to be a participant in this study

Print Name of Participant_____

Signature of Participant _____

Date _____
Day/month/year

Case study at SAP SE

Observational protocol for observing the estimation sessions of agile development teams at SAP SE

General instructions:

During the observation, the researchers/observers, would observe and document the sequence of events, and tasks performed by the participants at SAP. Participants include development teams, scrum master and product owner (see Table 1).

Table 1: Participants and abbreviations

Researchers/Observers	Development team1	Development team2
R1: Researcher 1	P: Person 1....	...
R2: Researcher 2		

The researchers would write down the tasks performed by the participants as well as the events in the time they occur. Events, include, e.g., starting a new task, questions, interruptions or unexpected behaviors/reactions.

For describing the sequence of tasks and events performed by the participants while doing the effort estimation session, please keep in mind the following questions:

- **What task** is being performed?
- **Who** participates in the task?
- **What information / criteria** is being used for making decisions?
- (If applicable) **What is the rationale** behind decisions?
- **What sorts of emotions** are being expressed by participants during this task? How can you be sure? (see Table 3)

All these data are very important to better understand the estimation process of change requests in the context of agile software development for establishing the current state of practice at SAP.

Table 2 contains examples about tasks, information, criteria, and Table 3 contains emotions to be collected in the observation protocol. The table does not provide a complete set of tasks and additional tasks, information, criteria, and emotions can be added or removed accordingly.

Table 2: Example of tasks and information used

On task	Passive task	Off task (or event)
<ul style="list-style-type: none"> – Analyzing change requests – Analyzing artifacts, e.g., user stories, historical data, code, and test cases – Selecting and prioritizing change requests – Estimating change requests – Outcomes of estimation i.e. documenting and communicating estimates 	<ul style="list-style-type: none"> – Thinking silently – Consulting artifacts, e.g., user stories, historical data, code and test cases 	<ul style="list-style-type: none"> – Participant(s) is (are) inactive – Participants talk about other topics – Participant arrives late – Participant leave early – Cell-phone call – Reviewing emails – Noise – Interruption

Information used	Criteria
<ul style="list-style-type: none"> – Artifacts, e.g., <ul style="list-style-type: none"> ○ User stories ○ Code ○ Test cases – Historical data – Individual experience 	<ul style="list-style-type: none"> – Developers experience – Change complexity/size – Change impact on existing system

Table 3: Example of emotions exhibited

<ul style="list-style-type: none"> - (Un) concentrated / - (Un) focused - (In) attentive - Thoughtful - (Un) confident - Doubtful - Clear 	<ul style="list-style-type: none"> - Satisfied - Bothered - Frustrated - Calm - Upset - (Un) worried - Confuse 	<ul style="list-style-type: none"> - (Un) Motivated - (Un) Interested - Excited - Bored - Relaxed - Stressed - Tired 	<ul style="list-style-type: none"> - Independent - Dependent - Autonomous - Open - Closed - Collaborative - Isolated
--	---	---	---

Observational protocol

Start time: _____ [hh:mm]

Observer: ☐ R1 ☐ R2 ☐ Other: _____

Stakeholder: Please specify here stakeholders' identifiers and names.

ID	Name

Sequence	Start Time	Description Who? When? What? Why? Information used? Emotions?

End time: _____ [hh:mm]

Case study at SAP SE

Interview questionnaire for establishing state-of-the-practice of effort estimation at SAP SE

Interview Goal: To understand and characterize the estimation process w.r.t. its effectiveness in the context of agile software development from the perspective of SAP agile development teams.

Interviews protocol: Interview questionnaire has three sections. First section is about demographic questions, second section is related to current estimation process and third section is to about the needs for improvement.

1 Demographic Information

1.1 Interviewee Demographics

Full name	
Contact information (email)	
Years of experience in software development	
Years of experience in agile development	
Current agile practice	<input type="checkbox"/> Scrum <input type="checkbox"/> Extreme programming ...
Role in current development	<input type="checkbox"/> Scrum Master <input type="checkbox"/> Product Owner <input type="checkbox"/> Developer <input type="checkbox"/> Architect <input type="checkbox"/> Quality manager <input type="checkbox"/> Other, please specify:

1.2 Product context:

Product name	
Product size	LOC, FP, features
Domain	<input type="checkbox"/> Information system <input type="checkbox"/> embedded system <input type="checkbox"/> web-application <input type="checkbox"/> distributed system
Use of computer aided software engineering (CASE) tools	<input type="checkbox"/> integrated development environment <input type="checkbox"/> automated test tools <input type="checkbox"/> estimation tools
Team size	
How many teams are involved in this product development?	
Organization size (business unit)	Number of employees
Teams are collocated or distributed?	
Your team type	<input type="checkbox"/> Cross functional <input type="checkbox"/> Dedicated <input type="checkbox"/> mix
How the teams are related, do you get any features from them to estimate and implement or you work independently?	
How many sprints have you delivered (how old is the product)?	<input type="checkbox"/> what % are bugs fixes, what % are enhancements
Programming language	

2 Current Estimation Process

The following set of questions aim at describing the current estimation process with respect to its scope, inputs, methods, and outputs. I would like to ask you a few questions regarding this very product development.

2.1 Estimation purpose

1. What is the abstraction level of estimation?

<input type="checkbox"/> Epic <input type="checkbox"/> Feature <input type="checkbox"/> User story	<input type="checkbox"/> Other, please specify:
--	---

2. Which of the attributes are estimated directly and which are derived from it?

<i>Directly estimated:</i> <input type="checkbox"/> Size <input type="checkbox"/> Effort/ cost <input type="checkbox"/> Schedule <input type="checkbox"/> Complexity <i>Derived from direct estimates:</i> <input type="checkbox"/> Size <input type="checkbox"/> Effort/ cost <input type="checkbox"/> Schedule <input type="checkbox"/> Complexity	<input type="checkbox"/> Other, please specify: e.g. size can be found from time and velocity maths etc. <input type="checkbox"/> Other, please specify:
---	--

3. What are the main purposes of doing estimation? Or to support what process this estimation is use for?

<i>Answers may include:</i> <input type="checkbox"/> Planning effort <input type="checkbox"/> Planning costs <input type="checkbox"/> Planning schedule	<input type="checkbox"/> Risk management <input type="checkbox"/> Finding impacted components <input type="checkbox"/> Finding dependencies <input type="checkbox"/> Other, please specify:
--	--

4. Who (roles) is involved in the estimation process (e.g. planning meetings)?

<input type="checkbox"/> Scrum master <input type="checkbox"/> Product owner <input type="checkbox"/> Development team <input type="checkbox"/> Quality manager	<input type="checkbox"/> Other, please specify:
--	---

5. How many team members are involved in estimating a single <abstraction level>?

Number of experts involved: ____

2.2 Estimation method

6. Which estimation methods are used to make estimates?

<p><i>Answer may include:</i></p> <ul style="list-style-type: none"> <input type="checkbox"/> Planning poker <input type="checkbox"/> Estimation game <input type="checkbox"/> Delphi method <input type="checkbox"/> Estimation by analogy <input type="checkbox"/> Model based estimation <input type="checkbox"/> Expert based estimation (one expert gives her estimate) 	<ul style="list-style-type: none"> <input type="checkbox"/> None <input type="checkbox"/> I don't know <input type="checkbox"/> Other, please specify:
--	---

2.3 Estimation inputs

7. What information do you need as input to this estimation method?

<p><i>Answer may include</i></p> <ul style="list-style-type: none"> <input type="checkbox"/> TbD: Option list <p><i>Developer's data:</i></p> <ul style="list-style-type: none"> <input type="checkbox"/> Developer's experience with assessment <input type="checkbox"/> Developer's knowledge of <abstraction level> <input type="checkbox"/> Dependencies among <abstraction level> <input type="checkbox"/> Developer's availability <input type="checkbox"/> Development type (bug fix, enhancement) <input type="checkbox"/> Developer's assessment (expert assessment) <p><i>Artifacts</i></p> <ul style="list-style-type: none"> <input type="checkbox"/> High level architecture <input type="checkbox"/> Reuse of existing artefacts <input type="checkbox"/> Required level of reusability of developed artefacts <input type="checkbox"/> How well the scope of < abstraction level > defined 	<p><i>Measurement data of already finished increments:</i></p> <ul style="list-style-type: none"> <input type="checkbox"/> Estimates made for a similar <abstraction level> in the previous completed sprints <input type="checkbox"/> <abstraction level> complexity <input type="checkbox"/> <abstraction level> dependency on other <abstraction level> <input type="checkbox"/> <abstraction level> impact on parts of the developed system <input type="checkbox"/> Time required to test <abstraction level> <input type="checkbox"/> Time required to test <abstraction level> + impacted parts <ul style="list-style-type: none"> <input type="checkbox"/> None <input type="checkbox"/> I don't know <input type="checkbox"/> Other, please specify:
--	--

8. *If the measurement data was used*, how much measurement data was available/used for estimation?

<ul style="list-style-type: none"> <input type="checkbox"/> Repository contained --- completed increments <input type="checkbox"/> Repository contained --- completed products 	<ul style="list-style-type: none"> <input type="checkbox"/> None <input type="checkbox"/> I don't know <input type="checkbox"/> Other, please specify:
--	---

9. *If the measurement data was used*, was measurement data complete?

10. Is this information stored? How? How is this information been collected?

Input/storage type/accessible to:

11. Is this information accessible to all team members?

12. How important do you consider these inputs for the estimation process?

	5: very important	4: important	3: not sure	2: not important	1: not important at all
<i>Option list (for each)</i>	○	○	○	○	○
<input type="checkbox"/> Developer's experience with assessment					
<input type="checkbox"/> Developer's knowledge of <abstraction level>					
<input type="checkbox"/> Developer's experience of implementation					
<input type="checkbox"/> Dependencies among <abstraction level>					
<input type="checkbox"/> <abstraction level> complexity					
<input type="checkbox"/> <abstraction level> impact on parts of the developed system					
<input type="checkbox"/> Estimates made for a similar <abstraction level> in the previous completed sprints					

13. To what extent this information can improve the estimation accuracy?

	5: Large extent	4: Certain extent	3: Not sure	2: Limited Extent	1: Not at all
<i>Option list(for each)</i>	○	○	○	○	○
<input type="checkbox"/> Developer's experience with assessment					
<input type="checkbox"/> Developer's knowledge of <abstraction level>					
<input type="checkbox"/> Developer's experience of implementation					
<input type="checkbox"/> Dependencies among <abstraction level>					
<input type="checkbox"/> <abstraction level> complexity					
<input type="checkbox"/> <abstraction level> impact on parts of the developed system					
<input type="checkbox"/> Estimates made for a similar <abstraction level> in the previous completed sprints					

2.4 Estimation process

14. What are the major steps of the estimation process? Please describe the major steps in the estimation process with respect to their inputs and outputs.

15. On what basis <abstraction level> are selected for a single sprint?

2.5 Estimation outcome

16. What is the unit of measure of this estimation?

☐ Story points, man-hours?

17. What are the outputs of the estimation method?

Answer may include

☐ Point estimate
☐ Interval estimate
☐ Most significant factors influencing estimate
☐ Dependencies between factors influencing estimate

☐ None
☐ I don't know
☐ Other, please specify:

18. What does story point reflects?

- a. Does it have a same perception among all the members?
- b. How do you come to real estimates (size, time or cost) with this metric?

2.6 Rework estimate

19. Do you rework estimates?

20. When in the development process do you rework?

21. Do you store all the estimates and re-estimates made during sprints? How?

3 Needs for improvement

The following set of questions aim at understanding your perception on the accuracy of the current estimation process. This will help us to identify needs for improvement.

3.1 Ease of use/Useful/ Ease to interpret

22. What are the benefits of these estimates? How are they used?

3.2 Accuracy

23. Using a scale from 5: very accurate to 1: not accurate at all, how accurate you think are the estimates made with the current estimation method?

5: Very accurate	4: Reasonable/ fairly accurate	3: Not sure	2: Less accurate	1: Not at all
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

(If answer to above question 2 or 1)

1. Do you consider that the estimates are usually over or under estimated?

2. What are the main reasons for this inaccuracy?

<p><i>Answer may include</i></p> <ul style="list-style-type: none"> <input type="checkbox"/> Choice of estimation method <input type="checkbox"/> <abstraction level> size <input type="checkbox"/> <abstraction level> complexity <input type="checkbox"/> Developer experience for making estimates <input type="checkbox"/> Lack of information (e.g. input data, impact of <abstraction level>, test cases) 		<ul style="list-style-type: none"> <input type="checkbox"/> Information was not considered (e.g. input data, impact of features, test cases) <input type="checkbox"/> Developer's availability <input type="checkbox"/> Expert bias <input type="checkbox"/> Lack of tool support <input type="checkbox"/> None <input type="checkbox"/> I don't know <input type="checkbox"/> Other, please specify:
--	--	--

- (If answer is = information was not available) What information was not available? Why?
- (If answer is = information was not considered) What information was not considered? Why?

3.3 Estimation bias

24. Does choice of method affect estimation bias?

25. Does size of user story affect estimation bias?

26. How estimation bias can be reduced?

3.4 Tool support

27. What aspect of the estimation process do you think should be (totally or partially) automated to better support the estimation process? *Present a list of optional to be sorted by the interviewee. For example:*

- Performing estimation
- Visualizing estimation
- Performing impact analysis
- Expert interaction and reworking estimates
- Re-visualization of estimates
- Projected estimates with some mean error
- Other: ...

3.5 Others

28. Is there any impact of estimation on the quality of product?

29. How do you consider non-functional requirements while doing estimation?

30. Is there any other aspect that you would like to change or improve in the current estimation process? Why?

31. Is there any aspect that you will not change in the current estimation process? Why?

B Appendix B

This appendix contains the questionnaire for establishing the state-of-the-practice at Insiders Technologies. The informed consent of Appendix A has been adapted and used.

Survey at Insiders Technologies

Questionnaire for establishing state-of-the-practice of effort estimation at Insiders Technologies

Goal: To understand and characterize the estimation process w.r.t. its effectiveness in the context of agile software development from the perspective of agile development teams.

Protocol: The questionnaire has seven sections. First section is about organization context. Second section is related to product characteristics and third is about project characteristics. Fourth deals with the estimation process and fifth is estimation context and measurement data. Sixth section is related to the impact related information. Seventh has the demographic questions.

1 Organisationaler Rahmen und Kontext

Frage	Antwort
1. Wie viele Mitarbeiter sind im Unternehmen beschäftigt?	Insgesamt: <bitte ergänzen> Direkt in der Softwareentwicklung: <bitte ergänzen>
2. In welche Domäne ist das Unternehmen einzuordnen?	<input type="checkbox"/> Informationssysteme <input type="checkbox"/> Eingebettete Systeme <input type="checkbox"/> Sonstige: <bitte ergänzen>
3. In welchen Marktsegmenten agiert das Unternehmen?	<input type="checkbox"/> Banken und Finanzdienstleistungen <input type="checkbox"/> Automobilindustrie <input type="checkbox"/> Landwirtschaft <input type="checkbox"/> Chemieindustrie <input type="checkbox"/> E-Commerce <input type="checkbox"/> Andere/weitere: <bitte ergänzen>
4. Welchem Geschäftstyp ist das Unternehmen zuzuordnen?	<input type="checkbox"/> Produkthersteller <input type="checkbox"/> Servicedienstleister <input type="checkbox"/> Sonstige: <bitte ergänzen>

2 Software Produktcharakteristika

Frage	Antwort
1. Welche Art von Produkten werden im Unternehmen hergestellt (bzw. welche Dienstleistungen werden angeboten)?	<input type="checkbox"/> Produkt-/Service-Portfolio <input type="checkbox"/> Kundenspezifisch <input type="checkbox"/> Andere/weitere: <bitte ergänzen>
2. Welcher Entwicklungsprozess wird bei der Planung und Umsetzung eines Produktes genutzt?	<input type="checkbox"/> Wasserfallmodell <input type="checkbox"/> Agile Entwicklung: <Welche: Scrum, XP, ...> <input type="checkbox"/> Organisationsspezifisch: <bitte in 2-3 Sätzen beschreiben>
3. Welche Programmiersprachen/Auszeichnungssprachen werden bei der Implementierung eines Produktes verwendet?	<input type="checkbox"/> Java <input type="checkbox"/> C# <input type="checkbox"/> C/C++ <input type="checkbox"/> HTML(5)/CSS(3) <input type="checkbox"/> PHP <input type="checkbox"/> XML <input type="checkbox"/> Sonstige/weitere: <bitte ergänzen>
4. Geben sie eine Einschätzung zum Umfang der umzusetzenden Produkte an, inklusive Maßeinheit (Kodezeilen, Story Points, usw.). Bsp., minimaler Umfang: 20 Story Points	Minimaler Umfang: Maximaler Umfang: Typischer Umfang:

3 Projektcharakteristika

Frage	Antwort
1. Wie viele Personen sind in einem Projekt involviert?	Minimal Teamgröße: Maximale Teamgröße: Typische Teamgröße:
2. Welche Rollen haben die beteiligten Personen inne?	<input type="checkbox"/> Projekt Manager <input type="checkbox"/> Produkt Manager/Owner <input type="checkbox"/> Quality Manager <input type="checkbox"/> Produkt Designer <input type="checkbox"/> Entwickler <input type="checkbox"/> Tester <input type="checkbox"/> Sonstige/weitere: <bitte ergänzen>

3. Wie lange dauert die Durchführung eines Projektes? Falls Projekt unterteilt wird (z.B. Iteration/Sprint, Release, usw.), bitte die Dauer für jede Granularitätsebene angeben.	Minimal Dauer: Maximale Dauer: Typische Dauer:
--	--

4 Schätzverfahren

Frage	Antwort
1. Welche Methoden werden zur Schätzung der Projektaufwände genutzt?	<input type="checkbox"/> Expertenschätzung <input type="checkbox"/> Daten-basierte algorithmische Verfahren <input type="checkbox"/> Kombination von experten- und datenbasierter Schätzung
2. Wie viele Personen sind in die Schätzung involviert?	
3. Welche Rollen haben die beteiligten Personen?	<input type="checkbox"/> Projekt Manager <input type="checkbox"/> Produkt Manager/Owner <input type="checkbox"/> Quality Manager <input type="checkbox"/> Produkt Designer <input type="checkbox"/> Entwickler <input type="checkbox"/> Tester <input type="checkbox"/> Sonstige/weitere: <bitte ergänzen>
a) Gibt es einen Verantwortlichen für die Schätzmethode?	<input type="checkbox"/> Ja: <Wer?> <input type="checkbox"/> Nein
b) Wer ist für die Durchzuführen der Schätzung im individuellen Projekt verantwortlich (Rolle)?	
4. Wann wird die Schätzung durchgeführt (z.B. nur zum Beginn des Projektes)?	<input type="checkbox"/> Akquise Phase <input type="checkbox"/> Planung des Projekts <input type="checkbox"/> Während der Projektlaufzeit <input type="checkbox"/> Sonstige: <bitte ergänzen>
5. Wie oft wird die Schätzung wiederholt bzw. aktualisiert?	<input type="checkbox"/> Nie <input type="checkbox"/> Bei jeder Iteration/Sprint <input type="checkbox"/> Bei jedem Release <input type="checkbox"/> Sonstige: <bitte ergänzen>
6. Werden am Ende die tatsächlichen Aufwände mit der originären Schätzung verglichen, z.B. um die Genauigkeit der Schätzungen zu bewerten?	<input type="checkbox"/> Immer <input type="checkbox"/> Oft <input type="checkbox"/> Selten <input type="checkbox"/> Nie
7. Welche Daten bzw. Informationsquellen werden für die Schätzung genutzt (z.B. Lastenheft, Stories)?	<input type="checkbox"/> Lastenheft <input type="checkbox"/> Aktuelle Kosten aus abgeschlossenen Projekten <input type="checkbox"/> Weitere: <bitte ergänzen>

5 Schätzkontext

5.1 Artefakte

Frage	Antwort
1. Wie werden die zur Schätzung benötigten Artefakte erfasst (z.B. in Form von Anforderungen)?	<input type="checkbox"/> Anforderungsspezifikation <input type="checkbox"/> Design <input type="checkbox"/> <u>Sonstige</u> : <bitte ergänzen>
2. Erfolgt eine Versionierung der Artefakte?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein
3. Wie oft werden die Artefakte während der Projektlaufzeit aktualisiert?	
4. Werden Werkzeuge zur Erfassung und Dokumentation der Artefakte verwendet? Wenn ja welche?	<input type="checkbox"/> Ja: <Welche> <input type="checkbox"/> Nein

5.2 Messdaten

Frage	Antwort																		
1. Welche produktbezogenen Messdaten werden erfasst? Welche Eigenschaften von welchen Produkten werden gemessen?	<p>Strukturelle Eigenschaften:</p> <ul style="list-style-type: none"><input type="checkbox"/> Anforderungsqualität<input type="checkbox"/> Code Größe<input type="checkbox"/> Code Komplexität<input type="checkbox"/> Architektur Compliance<input type="checkbox"/> <u>Weitere:</u> <Welche?> <p>Laufzeit Eigenschaften:</p> <ul style="list-style-type: none"><input type="checkbox"/> Testabdeckung<input type="checkbox"/> Software Antwortzeiten<input type="checkbox"/> Software Speicherverbrauch<input type="checkbox"/> Software Bugs Anzahl und Priorität<input type="checkbox"/> <u>Weitere:</u> <Welche?> <p>Sonstige Produkteigenschaften: <Welche?></p>																		
2. Welche projektbezogenen Messdaten werden erfasst?	<ul style="list-style-type: none"><input type="checkbox"/> Projektlaufzeit<input type="checkbox"/> Teamgröße<input type="checkbox"/> Teamerfahrung<input type="checkbox"/> Produktivität<input type="checkbox"/> Weitere: <Welche?>																		
3. Werden Projekte häufiger unter- oder überschätzt, d.h. sind die ursprünglichen Schätzungen kleiner oder größer als die tatsächlichen Werte nach Projektende?	<ul style="list-style-type: none"><input type="checkbox"/> Unterschätzt (ursprünglich < tatsächlich)<input type="checkbox"/> Überschätzt (ursprünglich > tatsächlich)<input type="checkbox"/> Gleich häufig unter- und überschätzt<input type="checkbox"/> Ich weiß nicht																		
4. Wie hoch ist die Differenz zwischen der ursprünglichen Schätzungen und der tatsächlichen Werten nach Projektende (relativ zum Schätzwert)? Unterscheiden sie bitte zwischen den Unterschätzung (ursprünglich < tatsächlich) und den Überschätzung (ursprünglich > tatsächlich)	<p>Aufwand</p> <table><tr><th>Abweichung</th><th>Unterschätzt</th><th>Überschätzt</th></tr><tr><td>Maximal</td><td>- %</td><td>+ %</td></tr><tr><td>Durchschnittlich</td><td>- %</td><td>+ %</td></tr></table> <p>Laufzeit</p> <table><tr><th>Abweichung</th><th>Unterschätzt</th><th>Überschätzt</th></tr><tr><td>Maximal</td><td>- %</td><td>+ %</td></tr><tr><td>Durchschnittlich</td><td>- %</td><td>+ %</td></tr></table>	Abweichung	Unterschätzt	Überschätzt	Maximal	- %	+ %	Durchschnittlich	- %	+ %	Abweichung	Unterschätzt	Überschätzt	Maximal	- %	+ %	Durchschnittlich	- %	+ %
Abweichung	Unterschätzt	Überschätzt																	
Maximal	- %	+ %																	
Durchschnittlich	- %	+ %																	
Abweichung	Unterschätzt	Überschätzt																	
Maximal	- %	+ %																	
Durchschnittlich	- %	+ %																	

	<p>Durchschnittliche Teamgröße</p> <table border="1"> <tr> <th>Abweichung</th><th>Unterschätzt</th><th>Überschätzt</th></tr> <tr> <td>Maximal</td><td>- %</td><td>+ %</td></tr> <tr> <td>Durchschnittlich</td><td>- %</td><td>+ %</td></tr> </table> <p>Produktumfang (LOC, Story Points, etc.)</p> <table border="1"> <tr> <th>Abweichung</th><th>Unterschätzt</th><th>Überschätzt</th></tr> <tr> <td>Maximal</td><td>- %</td><td>+ %</td></tr> <tr> <td>Durchschnittlich</td><td>- %</td><td>+ %</td></tr> </table>	Abweichung	Unterschätzt	Überschätzt	Maximal	- %	+ %	Durchschnittlich	- %	+ %	Abweichung	Unterschätzt	Überschätzt	Maximal	- %	+ %	Durchschnittlich	- %	+ %
Abweichung	Unterschätzt	Überschätzt																	
Maximal	- %	+ %																	
Durchschnittlich	- %	+ %																	
Abweichung	Unterschätzt	Überschätzt																	
Maximal	- %	+ %																	
Durchschnittlich	- %	+ %																	
5. Auf welcher Granularitätsebene wird die Schätzung durchgeführt? (z.B. Anforderung, Gesamtprojekt, Story)?																			
6. Für wie viele abgeschlossene Projekte, Releases, Versionen sind die Messdaten vorhanden?																			
7. Wie würden Sie die Zuverlässigkeit (Korrektheit) der vorhandenen Daten im Schnitt einschätzen?	<input type="checkbox"/> Sehr zuverlässig <input type="checkbox"/> Zuverlässig <input type="checkbox"/> Durchschnittlich zuverlässig <input type="checkbox"/> Wenig zuverlässig <input type="checkbox"/> Nicht zuverlässig																		
8. Wie würden Sie die Vollständigkeit der vorhandenen Daten (in %) im Schnitt einschätzen?	<input type="checkbox"/> 100 – 96% <input type="checkbox"/> 95 – 76% <input type="checkbox"/> 75 – 51% <input type="checkbox"/> 50 – 26 % <input type="checkbox"/> 25 – 0%																		
9. Beschreiben die vorhandenen Daten die üblichen Projekte, die vom Unternehmen abgewickelt werden?	<input type="checkbox"/> Ja <input type="checkbox"/> Nein (Daten sind nicht repräsentativ)																		

6 Impact analysis related

1. Which impact level is appropriate for effort estimation and why?

- ☐ Method
☐ Class
☐ Other

Reason for your answer: _____

2. Which way would you prefer to start impact analysis, and why?

- ☐ Indirect way – Formulating a query from description of change request (backlog item), then identifying potentially impacted methods/classes returned by the system.
☐ Direct way - Reading the change request (backlog item) description and identifying potentially impacted methods/classes directly in the code.
☐ Other:

Reason for your answer: _____

3. What is considered as impact factor?

Factor	Measured as		
<input type="checkbox"/> Size	<input type="checkbox"/> Total Lines of code of potentially impacted methods/classes	<input type="checkbox"/> No. of potentially impacted methods/classes	<input type="checkbox"/> Other:
<input type="checkbox"/> Complexity	<input type="checkbox"/> McCabe's Cyclomatic complexity (CC) for a method (No. of decision points/ control flow in a method)	<input type="checkbox"/> Halstead metrics (No. of operator/ operands)	<input type="checkbox"/> Other:
<input type="checkbox"/> Coupling	<input type="checkbox"/> Fan in (Fan-in of a method/class is the No. of methods/classes that depends directly on it)	<input type="checkbox"/> Fan out (Fan-out of a method/class is the No. of methods/classes that it depends on)	<input type="checkbox"/> Other:
<input type="checkbox"/> Frequency of change	<input type="checkbox"/> No. of times potentially impacted methods/classes are changed in version history		
<input type="checkbox"/> Percentage of change	<input type="checkbox"/> Ratio of No. of added/deleted lines to the total LOC of potentially impacted methods/classes in version history		
<input type="checkbox"/> Other			

4. Prioritize the following information according to their importance for effort estimates (where 1 stands for highest priority).

Impact factors	Importance for effort estimation
Size	
Complexity	
Coupling	
Frequency of change of the potentially impacted methods/classes in version history	
Percentage of change of the potentially impacted methods/classes in version history	
Other:	

5. How useful are the following impact factors for making effort estimates?

Impact factors	Useful	Not sure	Not useful
Size	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Complexity	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Coupling	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Frequency of change	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Percentage of change	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. How useful are the following coupling metrics for making effort estimates?

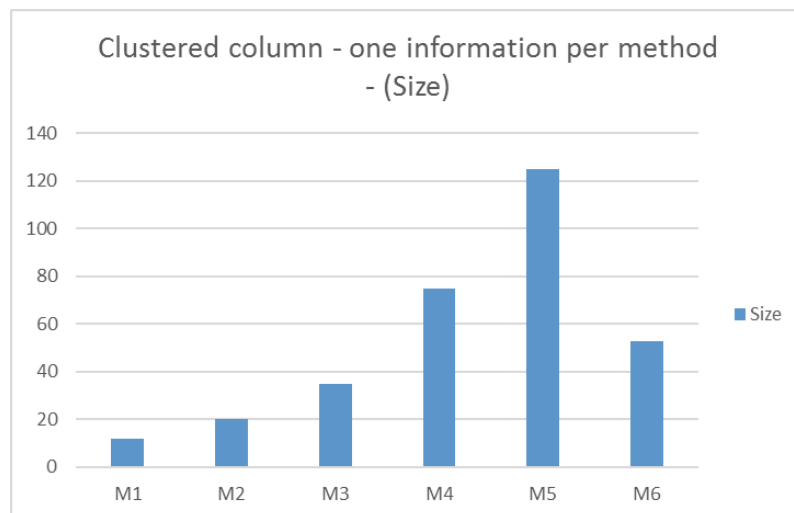
Coupling metrics	Useful	Not sure	Not useful
Cohesion	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fan-in	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fan-out	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
No. of children of a class	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Depth of inheritance tree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. Which information pair is the most relevant for effort estimation?

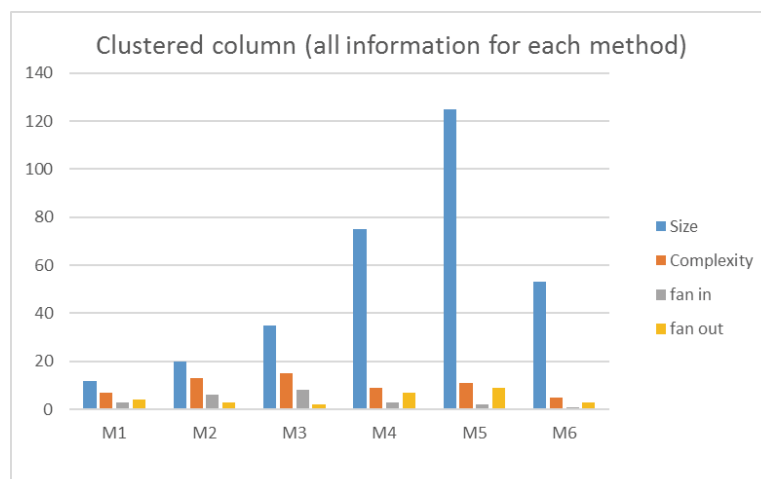
- ☐ Size – Complexity
- ☐ Size – Coupling
- ☐ Complexity – Coupling
- ☐ Other:

8. Which of the following visualizations are useful to show impact information?

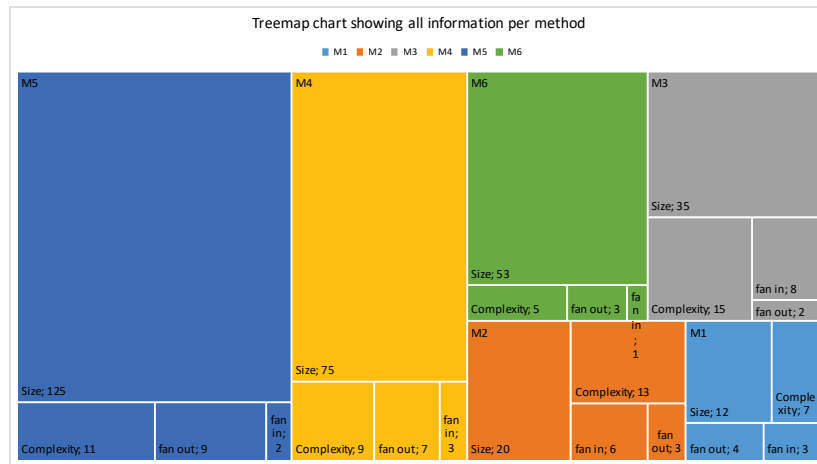
- ☐ Display one information in one chart. For example, this chart displays only “size” per method.



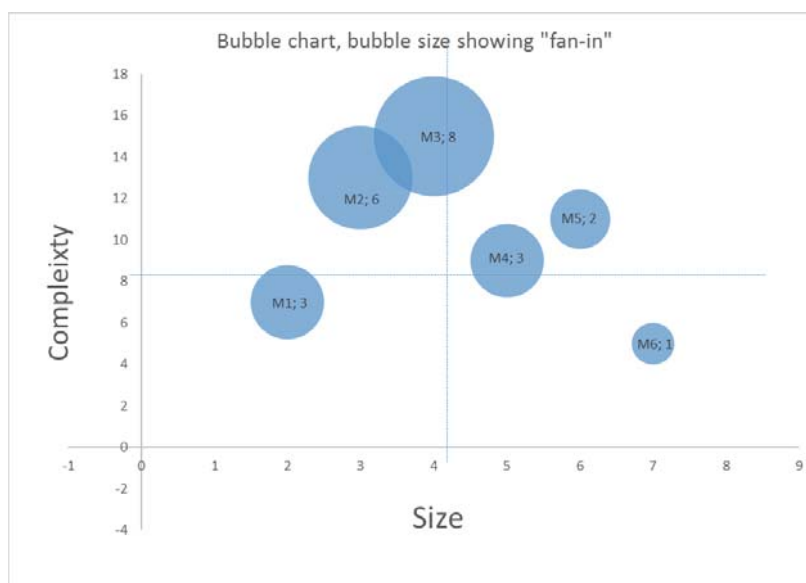
- ☐ Display all information in one chart. For example, this chart displays “size, complexity, fan-in and fan-out” for all methods



- ☐ Display all information in one chart. For example, this treemap chart displays “size, complexity, fan-in, fan-out” for all methods.



- ☐ Display maximum information in one chart. For example, this bubble chart displays “size, complexity” on x and y axis respectively where, “fan-in” is shown as bubble size.



9. Is the impact information and visualizations presented are sufficient to make effort estimate?

☐ Yes

☐ No

If no, please explain:

10. Do you require any support (e.g. point estimates- model based estimation method) for making estimates in addition to having impact information and visualizations?

☐ Yes

☐ No

If yes, please explain the reason:

7 Contact data

Name	
Email	
Role	

C Appendix C

This appendix contains the informed consent and the questionnaire for evaluation at SAP SE.

Informed content for evaluation at SAP SE

Informed Consent Form for development teams at SAP who we are inviting to participate in the evaluation workshop. The purpose of this workshop is to evaluate the usefulness of HyEEASe and the mockup to support effort estimation in agile software development.”

This Informed Consent Form has two parts:

- Information Sheet (to share information about the workshop with you)
- Certificate of Consent (for signatures if you choose to participate)

You will be given a copy of the full Informed Consent Form

Who to Contact

If you have any questions you may ask them now or later, even after the study has started. If you wish to ask questions later, you may contact any of the following researchers: Binish Tanveer (binish.tanveer@iese.fraunhofer.de)

Part I: Information Sheet

Introduction

I am Binish Tanveer, working in the Process Engineering department at Fraunhofer IESE. I am doing research on improving the process of effort estimation in agile software development methodologies. In particular, I am currently developing a conceptual framework for improving effort estimation in agile software development. A part of this research is being conducted in the context of Software Campus project initiative, funded by BMBF grant number 01IS12053, titled "HyEEASe – Hybrid Effort Estimation of increments in Agile and Incremental Software Development". I am going to give you information and invite you to be part of this research.

Context:

Effort estimation is more challenging in an agile context, as instead of exerting strict control over changes in requirements, dynamism is embraced. Current practice relies on expert judgment, where the accuracy of estimates is sensitive to the expertise of practitioners and prone to bias.

In order to improve the effectiveness of the effort estimation process, we investigated the estimation process with respect to its accuracy in the context of agile software development from the perspective of agile development teams.

Using case study research, two observations and eleven interviews were conducted with three agile development teams at SAP SE.

The results reveal that factors such as the developer's knowledge, the complexity and impact of changes on the underlying system affect the magnitude as well as estimation accuracy. Furthermore, there is a need for a tool that incorporates expert knowledge, enables explicit consideration impact information and visualizes this information in order to improve the effectiveness of the effort estimation.

Based on the findings of the case study, a conceptual framework, inspired by the Quality Improvement Paradigm is proposed to improve effort estimation in agile development by utilizing impact analysis. Moreover, a mockup has been designed to support the instantiation of this conceptual framework. It shows the workflow of how impact analysis will be utilized and support effort estimation in agile development context.

Purpose of this workshop:

Consequently, the **purpose of this workshop** is to:

Analyze HyEEASe and the mock up
for the purpose to *evaluate with respect to the usefulness*
from the perspective of agile development teams
in the context of estimation in agile software development.

Analyze HyEEASe and the mock up
for the purpose of *identifying needs for improvements* to increase their quality
from the perspective of agile development teams
in the context of agile software development.

We believe that you can help us to better understand what and how impact information can be utilized to support estimation of effort of implementing change requests. Moreover, the feedback regarding strength and weaknesses of HyEEASe or mockup as well as needs for improving them will also be very helpful in furthering the research.

Participant Selection

You are being invited to take part in this research because we feel that your experience as agile software developer, tester or product owner can contribute to our understanding of utilizing change impact analysis for effort estimation in agile. Your participation in this evaluation workshop is entirely voluntary.

Feedback of the evaluation will be provided to SAP as soon as the analysis of the evaluation is completed.

Workshop procedure

The evaluation workshop is designed for two hours duration. The instrument used for evaluation is a questionnaire. All participants should fill out different questionnaires, one for HyEEASE and other for mockup evaluation.

The workshop includes the following steps:

1. Introduction to the research and purpose of workshop
2. Evaluation of HyEEASE
3. Introduction to the mockup
4. Evaluation of the mockup
5. Further feedback

During these steps, all participants and the researchers will be located in the same room. Please fill out the questionnaires as much as possible. The researchers will support you and answer questions anytime if necessary.

Confidentiality

Only the researchers will have access to the information that will be collected from this workshop and it will be kept confidential. Personal information about you and your given answers will be put away and only the researchers will be able to see them. Any information about you will have an identification number instead of your name. No-one but the researchers will know what your number is and we will keep that information under lock and key. It will not be shared with or given to anyone except the research team.

Sharing the Results

1. **Information** (i.e. raw data that is collected directly from this workshop)
Nothing that you tell us will be shared with anybody outside the research team even inside SAP and the participants of this research, and nothing will be attributed to you by name.
2. **Knowledge** (i.e. the aggregated results of evaluation workshop)
The knowledge that we get from this workshop will be shared with you and the other participants before it is made widely available to the public. Each participant will receive a summary of the results. This knowledge will be published as part of Binish Tanveer's PhD Thesis "Utilizing change impact analysis for effort estimation in agile software development". In addition, we will publish the anonymized results in order that other interested people may learn from our research e.g. conferences, journals.

Right to refuse or withdraw

Participation in this study is entirely voluntary. It is your choice whether to participate or not. You do not have to take part in this research if you do not wish to do so, and choosing to participate will not affect your job or job-related evaluations in any way. You may stop participating in the study at any time that you wish without your job being affected.

Part II: Certificate of Consent

I have been invited to participate in evaluation workshop about utilizing impact analysis for effort estimation in agile software development.

I have read the foregoing information, or it has been read to me. I have had the opportunity to ask questions about it and any questions I have asked have been answered to my satisfaction. I consent voluntarily to be a participant in this workshop.

Print Name of Participant _____

Signature of Participant _____

Date _____
Day/month/year

Questionnaire - HyEEASe

Date: 16.02.2017

Evaluation at SAP SE

Purpose of this workshop:

The **purpose of this workshop** is to:

Analyze HyEEASe and the mock up
for the purpose to *evaluate with respect to the understandability and usefulness*
from the perspective of agile development teams
in the context of estimation in agile software development.

Analyze HyEEASe and the mock up
for the purpose of *identifying needs for improvements* to increase their completeness and usefulness
from the perspective of agile development teams
in the context of agile software development.

Questionnaire - HyEEASE

Date: 16.02.2017

Workshop procedure

The evaluation workshop is designed for two hours duration. The instrument used for evaluation is a questionnaire. All participants should fill out different questionnaires, one for HyEEASE evaluation and other for mockup evaluation.

The workshop includes the following steps:

1. Introduction to the research and purpose of workshop

In a presentation, a quick introduction of the research will be conducted explaining the purpose of the workshop as well as introducing the conceptual framework and ideas of its instantiation. Open questions regarding the evaluation will be answered. Informed consent will be collected.

2. Evaluation HyEEASE

This step consists of evaluating HyEEASE and ideas presented in the previous presentation. Each participant shall evaluate HyEEASE overall using a questionnaire that will be given to them. The questionnaire also has four demographic questions. After this step, any of your open questions will be answered before continuing with the next step.

3. Introduction to the mockup

This step consists of a walkthrough including a short presentation and explanation of the mockup. Open questions regarding the evaluation will be answered. After this walkthrough, any of your open questions will be answered before continuing with the next evaluation step.

4. Evaluation of the mockup

Based on the previous presentation and walkthrough, every participant will receive another questionnaire to evaluate different quality aspects of the mockup.

5. Further feedback

In the end of this study, the researchers will ask you about further comments regarding HyEEASE and/or the mockup and you can provide your feedback.

During these steps, you and the researchers will be located in the same room. Please fill out the questionnaires as much as possible. The researchers will support you and answer questions anytime if necessary.

Document information

This document contains the questionnaires to evaluate HyEEASE and mockup presented earlier. The first questionnaire contains demographic questions as well as questions evaluating HyEEASE overall with respect to its *understandability and usefulness*. The second questionnaire contains questions for evaluating the quality aspects of the mockup. The questionnaires will be provided separately in two sessions.

Contact information

Binish Tanveer (main researcher)

binish.tanveer@iese.fraunhofer.de

Questionnaire - HyEEASe

Date: 16.02.2017

Demographic Information

Please provide the following information

Full name:	
Years of experience in software development:	
Years of experience in agile development:	
Current role in team:	<input type="checkbox"/> Scrum Master <input type="checkbox"/> Product Owner <input type="checkbox"/> Developer <input type="checkbox"/> Architect <input type="checkbox"/> Quality manager <input type="checkbox"/> Other, please specify: _____

Questionnaire - HyEEASe

Date: 16.02.2017

Questionnaire – HyEEASe

Your **Full** **Name:**

Please answer the following questionnaire about HyEEASe method that utilizes change impact analysis for supporting effort estimation. To what degree you agree or disagree with the following statements:

1.

HyEEASe is	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
...clear in meaning.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...easy to comprehend.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In general, it is understandable.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2.

HyEEASe is	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
...informative to my work.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...valuable to my work.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...applicable to my work.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In general, it is useful for estimating efforts.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. In your opinion, what information or steps are missing in the HyEEASe?

- ☐ No
- ☐ Yes, please briefly explain:

4. What information do you consider as indicator of impact?

Questionnaire - Mockup

Date: 16.02.2017

Questionnaire – Mockup

Your Full Name:

This questionnaire consists of **12 questions** concerning the **understandability**, **usefulness**, and **relevance** of a potential tool presented through the mockup. The questions regarding the **contained information** and potential **visualizations** in the mockup are also part of this questionnaire.

Based on the presented mockup as a potential tool for supporting effort estimation in software development, to what degree you agree or disagree with the following statements:

1.

Assuming the mockup will be implemented as a tool, I expect its usage:	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
...to be clear.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...to be understandable.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...requires a lot of my mental effort.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...to be easy to use.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2.

Assuming the mockup will be implemented as a tool, I expect it to be:	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
...informative to my work.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...valuable to my work.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...is appropriate for my work.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...is applicable to my work.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In general, it is useful for estimating efforts.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3.

Assuming the mockup will be implemented as a tool:	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
I intend to use it.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I predict that I would use it.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I plan to use the system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Questionnaire - Mockup

Date: 16.02.2017

4. In the potential tool, which of the following information is important to consider as impact indicators? (multiple answers possible)

- ☐ *Total Size* of the potentially impacted methods/classes (measured in lines of code-LOC)
- ☐ *Complexity* of the potentially impacted methods/classes (measured as McCabe's Cyclomatic complexity (CC) for a method and average McCabe's cyclomatic complexity (AMC) for a class)
- ☐ *Coupling* of the potentially impacted methods/classes
- ☐ *Frequency of change* of the potentially impacted methods/classes in version history
- ☐ *Percentage of change* of the potentially impacted methods/classes in version history
- ☐ *Other*, please specify:

5. In the potential tool, which impact level is appropriate for effort estimation? Please also explain the reason for your choice.

- ☐ Class Level
- ☐ Method Level
- ☐ Other, please specify:

Reason, please specify:

Questionnaire - Mockup

Date: 16.02.2017

6. In the potential tool, how useful will be the following information for effort estimation?

	1: not at all useful	2: not very useful	3: neutral	4: somewh at useful	5: very useful	I don't know
<i>Total Size</i> of the potentially impacted methods/classes (LOC)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Complexity</i> of the potentially impacted methods/classes (CC or AMC)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Coupling</i> of the potentially impacted methods/classes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Frequency of change</i> of the potentially impacted methods/classes in version history	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Percentage of change</i> of the potentially impacted methods/classes in version history	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other: _____	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. In the potential tool, how useful will be the following coupling metrics for effort estimation?

	1: not at all useful	2: not very useful	3: neutral	4: somewh at useful	5: very useful	I don't know
<i>Cohesion</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Fan-in:</i> Fan-in of a method/class is the number of methods/classes that depends directly on it	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Fan-out:</i> Fan-out of a method/class the number of methods/classes that it depends on	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Number of children for a class</i> i.e. the number of its sub- classes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>Depth of inheritance tree for a class</i> i.e. the number of its base classes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other: _____	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Questionnaire - Mockup

Date: 16.02.2017

8. Please prioritize the following information according to their importance for effort estimates (where 1 stands for highest priority).

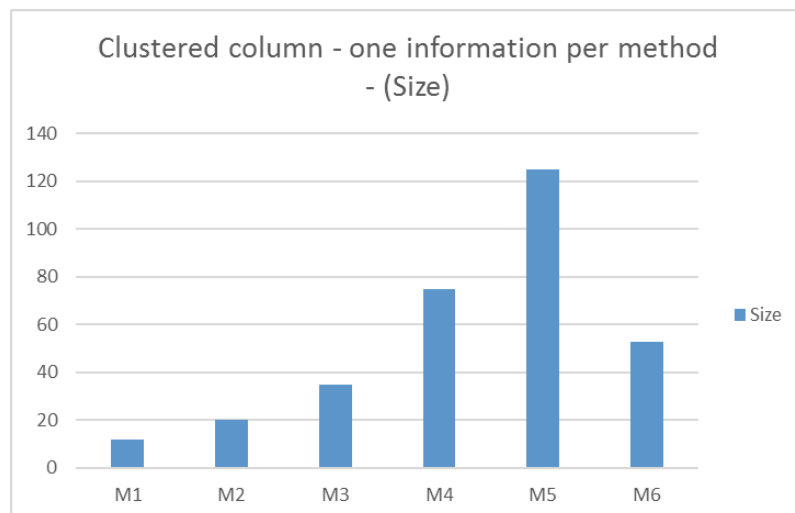
	Priority
<i>Total Size</i> of the potentially impacted methods/classes (LOC)	_____
<i>Complexity</i> of the potentially impacted methods/classes (CC or AMC)	_____
<i>Coupling</i> of the potentially impacted methods/classes	_____
<i>Frequency of change</i> of the potentially impacted methods/classes in version history	_____
<i>Percentage of change</i> of the potentially impacted methods/classes in version history	_____
Other: _____	_____

Questionnaire - Mockup

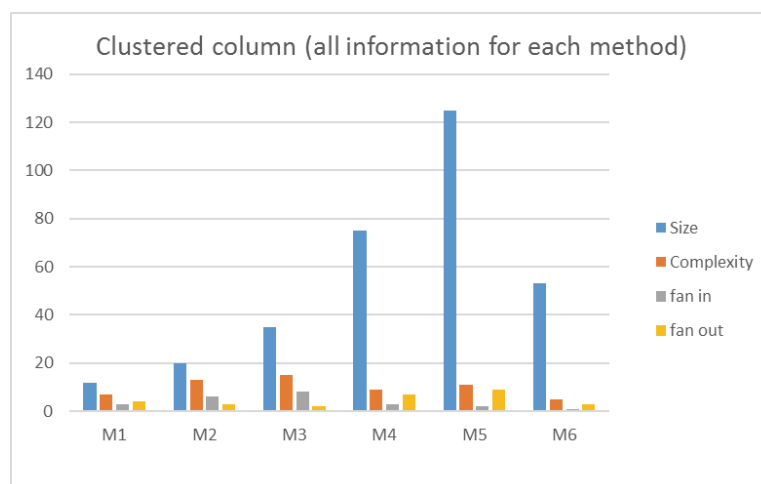
Date: 16.02.2017

9. Which of the following four visualizations are useful to show the impact information? Select all those that are useful.

- ☐ Display one information in one chart. For example, this chart displays only “size” per method.



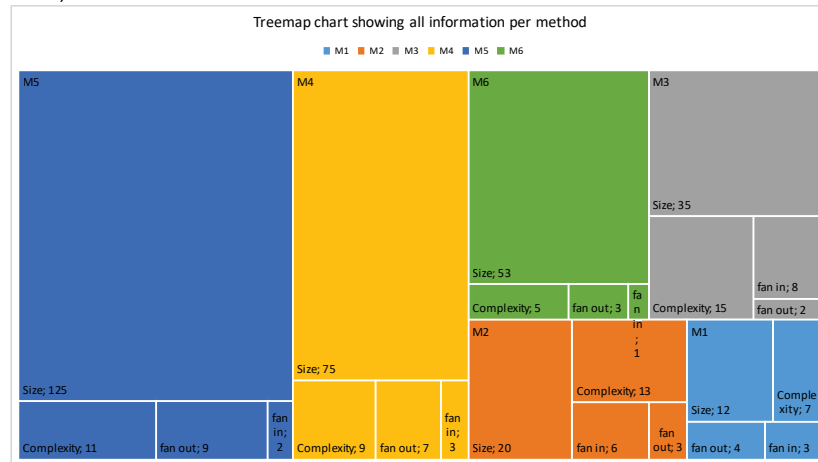
- ☐ Display all information in one chart. For example, this chart displays “size, complexity, fan-in and fan-out” for all methods



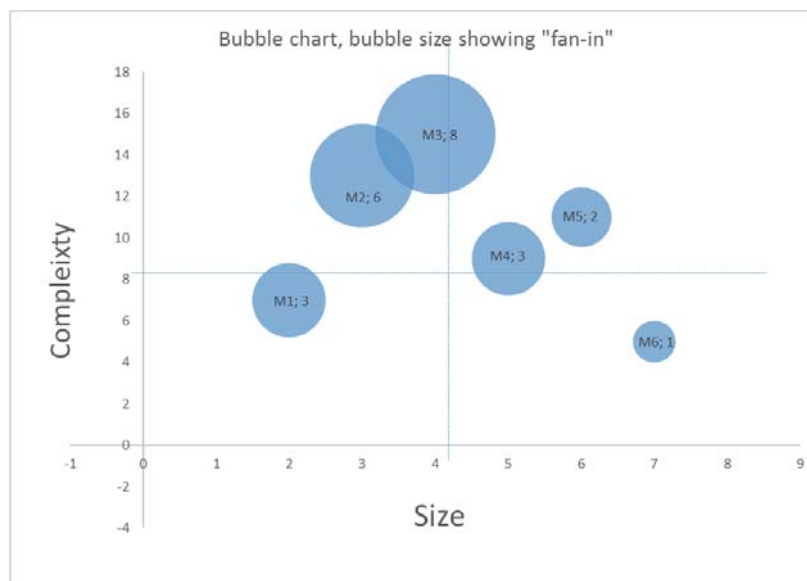
Questionnaire - Mockup

Date: 16.02.2017

- Display all information in one chart. For example, this treemap chart displays “size, complexity, fan-in, fan-out” for all methods.



- Display maximum information in one chart. For example, this bubble chart displays “size, complexity” on x and y axis respectively where, “fan-in” is shown as bubble size.



- Other: If you think of any other visualization, please briefly describe

Questionnaire - Mockup

Date: 16.02.2017

10. For the potential tool, which information pair is the most relevant for effort estimation? (only one answer possible)

- ☐ Size – Complexity
- ☐ Size – Coupling
- ☐ Complexity – Coupling
- ☐ Other, please specify:

11. Is the impact information and visualizations presented in the mockup sufficient to support effort estimate?

- ☐ Yes
- ☐ No, please specify what other information is required or what other step should be performed to make effort estimate?

12. If you have any other comments, please provide here:

Questionnaire - Mockup

Date: 16.02.2017

Overall Feedback

In the following part, we are interested in your further **feedback and suggestions for improvement regarding the presented mockup**. Please answer the next questions by using the provided moderation cards. Please write

- in **block letters**
 - only **one issue/subject per card**
 - maximum **3 lines per code**
1. On the **yellow cards**, write these aspects of HyEEASe and the presented mockup that **you liked the most**.
 2. On the **red cards**, write these aspects of HyEEASe and the presented mockup that **should be improved or** those who **represent a problem/risk** for their usefulness, applicability and/or supporting experts in making effort estimates.

Please return your written cards after completion.

Thank you very much!

D Appendix D

This appendix contains the informed consent and the questionnaire for evaluation at Insiders Technologies.

Informed content for evaluation at Insiders Technologies

Informed Consent Form for effort estimation experts/teams at Insiders Technologies, one of the members of Abakus SMEs who we are inviting to participate in the evaluation workshop. The purpose of this workshop is to

“Analyze the HyEEASe (Hybrid Effort Estimation of changes in Agile Software development) method and the prototype for the purpose to evaluate with respect to the *understandability and usefulness* as well as *identifying needs for improvements* from the perspective of agile development teams in the context of estimation in agile software development.”

This Informed Consent Form has two parts:

- Information Sheet (to share information about the workshop with you)
- Certificate of Consent (for signatures if you choose to participate)

You will be given a copy of the full Informed Consent Form

Who to Contact

If you have any questions you may ask them now or later, even after the study has started. If you wish to ask questions later, you may contact: Binish Tanveer (binish.tanveer@iese.fraunhofer.de).

Part I: Information Sheet

Introduction

I am Binish Tanveer, working in the Process Engineering department at Fraunhofer IESE. I am doing research on improving the process of effort estimation in agile software development methodologies. In particular, I am currently developing a hybrid approach for improving effort estimation in agile software development. A part of this research is being conducted in the context of Abakus project, grant number 01IS15050G. I am going to give you information and invite you to be part of this research.

If you have questions, you can contact me as.

Context:

Effort estimation is more challenging in an agile context, as instead of exerting strict control over changes in requirements, dynamism is embraced. Current practice relies on expert judgment, where the accuracy of estimates is sensitive to the expertise of practitioners and prone to bias.

In order to improve the effectiveness of the effort estimation process, we investigated the estimation process with respect to its accuracy in the context of agile software development from the perspective of agile development teams.

Using case study research, two observations and eleven interviews were conducted in a Germany based multinational software company with three agile development teams.

The results reveal that factors such as the developer's knowledge, the complexity and impact of changes on the underlying system affect the magnitude as well as estimation accuracy. Furthermore, there is a need for a tool that incorporates expert knowledge, enables explicit consideration impact information and visualizes this information in order to improve the effectiveness of the effort estimation.

Based on the findings of the case study, a hybrid estimation approach based on utilizing change impact analysis and supported by a prototype is developed to improve effort estimation in agile development.

Purpose of this workshop:

Consequently, the **purpose of this workshop** is to:

Analyze the HyEEASe method and prototype tool
for the purpose to *evaluate with respect to the understandability and usefulness*
from the perspective of agile development teams
in the context of estimation in agile software development.

Analyze the HyEEASe method and prototype tool
for the purpose of *identifying needs for improvements* to increase their completeness and usefulness
from the perspective of agile development teams
in the context of agile software development.

Participant Selection

You are being invited to take part in this research because we feel that your experience as effort estimation experts in agile software development (being developer, tester or product owner) can contribute much to improve the HyEEASe method and prototype. Your participation in this workshop is entirely voluntary.

Workshop procedure

The evaluation workshop takes place on 13th Nov 2017 at Insiders Technology and is designed for four hours duration starting 14:00 till 18:00. The instrument used for evaluation is a questionnaire. All the participants should fill out this questionnaire. The researchers will answer open questions, if any, during these steps.

The workshop includes the following steps:

1. Introduction and purpose of workshop

In a presentation, a quick introduction of the research will be conducted explaining the purpose of the workshop as well as introducing the HyEEASe method and prototype. Informed consent will be collected.

2. Training of the prototype

This step consists of a walkthrough including a short presentation and explanation of the prototype. You will be asked if you feel prepared to use the prototype tool (with the researcher) before proceeding with the next step.

3. Using the prototype to estimate the tasks

Together with the researcher, you will be able to use the prototype tool and make estimations for the selected set of tasks. You will make estimates using your current estimation method. You are free to make discussions with other participants while making estimation. The researchers will be noting down your individual estimates as well.

4. Evaluation of the method and prototype

This step consist of evaluating the HyEEASe method and prototype. Each one of you shall evaluate the method and prototype using a questionnaire that will be given to you. The questionnaire also have four demographic questions.

5. Further feedback

In the end of this study, the researchers will ask you about further comments regarding the method and prototype and you can provide your feedback. The feedback will be collected as follows:

- You will be provided with the moderation cards in green and red colors.
- Please write
 - in **block letters**
 - only **one issue/subject per card (max issue/subject = three)**
 - maximum **3 lines per card**
- On the **green cards**, write the aspects of the method and the prototype that **you liked the most**.
- On the **orange cards**, write the aspects of the the method and the prototype that **should be improved or** those who **represent a problem/risk** regarding their usefulness, applicability and/or supporting experts in making effort estimates.
- Please return your written cards after completion.

During these steps, you and the researchers will be located in the same room. Please fill out the questionnaires as much as possible. The researchers will support you and answer questions anytime if necessary.

Document information

This document contains the questionnaires to evaluate the method and prototype presented earlier. The first questionnaire contains demographic questions as well as questions evaluating the quality of the method and prototype with respect to its *understandability and usefulness*.

Confidentiality

Only the researchers will have access to the information that will be collected from this workshop and it will be kept confidential. Personal information about you and your given answers will be put away and only the researchers will be able to see them. Any information about you will have an identification number instead of your name. No-one but the researchers

will know what your number is and we will keep that information under lock and key. It will not be shared with or given to anyone except the research team.

Sharing the Results

1. **Information** (i.e. raw data that is collected directly from this workshop)
Nothing that you tell us will be shared with anybody outside the research team even inside Insiders Technologies and Abakus participants, and nothing will be attributed to you by name.
2. **Knowledge** (i.e. the aggregated results of the workshop)
The knowledge that we get from this workshop will be shared with you and the other participants before it is made widely available to the public. Each participant will receive a summary of the results. This knowledge will be published as part of Binish Tanveer's PhD Thesis "Utilizing change impact analysis for effort estimation in agile software development". In addition, we will publish the anonymized results so that other interested people may learn from our research e.g. conferences, journals.

Right to refuse or withdraw

Participation in this study is entirely voluntary. It is your choice whether to participate or not. You do not have to take part in this research if you do not wish to do so, and choosing to participate will not affect your job or job-related evaluations in any way. You may stop participating in the study at any time that you wish without your job being affected.

Part II: Certificate of Consent

I have been invited to participate in evaluation workshop about analyzing the HyEEASe method and prototype for the purpose to *evaluate with respect to the understandability and usefulness* from the perspective of agile development teams at Insiders Technologies in the context of effort estimation in Abakus project.

I have read the foregoing information, or it has been read to me. I have had the opportunity to ask questions about it and any questions I have asked have been answered to my satisfaction. I consent voluntarily to be a participant in this workshop.

Print Name of Participant _____

Signature of Participant _____

Date _____
Day/month/year

Evaluation questionnaire

Date: 13th Nov 2017

Evaluation at Insiders Technologies

The **purpose of this workshop** is to:

Analyze the HyEEASe method and prototype tool
for the purpose to evaluate with respect to the *understandability and usefulness*
from the perspective of agile development teams
in the context of estimation in agile software development.

Analyze the HyEEASe method and prototype tool
for the purpose of identifying needs for *improvements to increase their completeness and usefulness*
from the perspective of agile development teams
in the context of agile software development.

Document information

This document contains the questionnaire to evaluate the method and prototype presented earlier.
The first part contains demographic questions whereas the second part contains questions regarding *understandability and usefulness* of the method and prototype.

Evaluation questionnaire

Date: 13th Nov 2017**Demographic Information**

Please provide the following information

Participant ID	HyEEASe008
Years of experience in agile development	
Years of experience in effort estimation	
Current effort estimation method	
Current role in team	<input type="checkbox"/> Scrum Master <input type="checkbox"/> Product Owner <input type="checkbox"/> Developer <input type="checkbox"/> Architect <input type="checkbox"/> Quality Manager <input type="checkbox"/> Other, please specify: _____

Evaluation questionnaire

Date: 13th Nov 2017

Please answer the following questionnaire about HyEEASe method and prototype for supporting effort estimation. To what degree you agree or disagree with the following statements:

1. I think the HyEEASe method...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... improves my performance in my job	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... increases my productivity in my job	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... enhances my effectiveness in my job	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is useful for my job	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. I think the HyEEASe method...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... is clear	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... requires a lot of mental effort to apply it	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is easy to use	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...does what I expected it to do	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. I think the HyEEASe method...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... is pretty much what I need to estimate effort of a task	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is missing critical data/information that would be very useful to estimate efforts of a task	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In general, is complete	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Evaluation questionnaire

Date: 13th Nov 2017

4. I think the information provided by HyEEASe when performing effort estimation ...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... includes <i>all relevant elements</i> required for estimating effort for new tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... has <i>sufficiently breath</i> for estimating effort for new tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... has an <i>appropriate level of detail</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In general, is <i>complete</i> for estimating effort for new tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. I think the information provided by the HyEEASe method is ...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... <i>clear in meaning</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... <i>easy to comprehend</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... <i>easy to interpret</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In general, <i>understandable</i> for estimating effort for new tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. I think the information provided by HyEEASe method is ...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... <i>trustworthy</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... <i>accurate</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... <i>credible</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In general, <i>reliable</i> for estimating effort for new tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. I think the information provided by HyEEASe method is ...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... <i>informative</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... <i>valuable</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In general, <i>useful</i> for estimating effort for new tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... <i>relevant</i> for estimating effort for new tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... <i>appropriate</i> for estimating effort for new tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... <i>applicable</i> for estimating effort for new tasks	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Evaluation questionnaire

Date: 13th Nov 2017

8. After using the HyEEASe method...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
I have sufficient knowledge about the potential impact of new tasks on our system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am more aware of potential impact of implementing a new task on our system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I know which are the most relevant impact factors.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I know the relative priority of impact factors.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9. I think the HyEEASe prototype ...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... is well formatted	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is well laid out	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is clearly presented on the screen	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... provides visualizations that support me doing effort estimations for new tasks						

10. Assuming I would have access to the HyEEASe prototype:	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
I intend to use it.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I predict that I would use it.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I plan to use the system.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

E Appendix E

E.1 COCOMO II Reuse model

For maintenance projects that involve more than 20% change in the existing base code (relative to new code being developed), COCOMO II uses maintenance size. An initial maintenance size is obtained in one of the two ways [34]:

1. When the base code size is known and the percentage of change to the base code is known i.e.

$$Size_m = [BaseCodeSize * MCF] * MAF$$

where MCF stands for Maintenance Change Factor (MCF) i.e. the percentage of change to the base code i.e.

$$MCF = \frac{SizeAdded + SizeModified}{BaseCodeSize}$$

2. When the fraction of code added or modified to the existing base code during the maintenance period is known. The deleted code is not counted i.e.

$$Size_m = [SizeAdded * SizeModified] * MAF$$

where MAF stands for Maintenance Adjustment Factor with which initial maintenance size estimate is adjusted i.e.

$$MAF = 1 + \left(\frac{SU}{100} * UNFM \right)$$

where SU is Software Understanding, and UNFM is Programmer Unfamiliarity. COCOMO II uses SU and UNFM factors from its reuse model to model the effects of well or poorly structured/understandable software on maintenance effort. The resulting maintenance effort estimation formula is the same as the COCOMO II Post-Architecture development model i.e. maintenance effort in person months (PM):

$$PM_m = A * \left(Size_m \right)^B * \prod_{i=1}^{17} EM_i$$

It uses the same parameters as the Post Architecture model with few exceptions and/or changes. For example, among scale factors, maintenance model does not use Required Development Schedule (SCED)

scale factor, similarly among cost drivers it does not use Developed for reusability (RUSE) and it uses a different rating scale for Required Software Reliability (RELY).

E.2 Set up of Agile COCOMO II - scale factors and cost drivers

Table E.1: Agile COCOMO II Scale factors ratings

	Very low	Low	Normal	High	Very high	Extra high	COCOMO multiplier
Precedentedness (PREC)	0	0	1	0	0	0	3.72
Development Flexibility (FLEX)	0	0	1	0	0	0	3.04
Architecture/ Risk resolution (RESL)	0	0	0	1	0	0	2.83
Team Cohesion (TEAM)	0	0	0	0	1	0	1.10
Process Maturity (PMAT)	0	0	0	1	0	0	3.12
Sum of scale factors (SF)	-	-	-	-	-	-	13.81

Table E.2: Agile COCOMO II Cost drivers ratings

	Task 1	Task 2	Task 3	Task 4	Multiplier
Task Complexity (CPLX)					
Level 1	0	0	0	0	0.73
Level 2	0	0	0	1	0.87
Level 3	0	1	1	0	1.00
Level 4	0	0	0	0	1.17
Level 5	1	0	0	0	1.34
Level 6	0	0	0	0	1.74
Database size (DATA)					
DB bytes / program SLOC < 10	0	0	0	0	0.90
10 ≤ D/P < 100	1	1	1	1	1.00
100 ≤ D/P < 1000	0	0	0	0	1.14
D/P ≥ 1000	0	0	0	0	1.28
Execution time Constraint (TIME)					
≤ 50% use of available execution time	1	1	1	1	1.00
70% use of available execution time	0	0	0	0	1.11
85% use of available execution time	0	0	0	0	1.29
95% use of available execution time	0	0	0	0	1.63
Main storage Constraint (STOR)					
≤ 50% use of available execution time	1	1	1	1	1.00
70% use of available execution time	0	0	0	0	1.05
85% use of available execution time	0	0	0	0	1.17

Continued on next page

Table E.2 – continued from previous page

	Task 1	Task 2	Task 3	Task 4	Multiplier
95% use of available execution time	0	0	0	0	1.46
Analyst Capability (ACAP)					
15th percentile	0	0	0	0	1.42
35th percentile	0	0	0	0	1.19
55th percentile	1	1	1	1	1.00
75th percentile	0	0	0	0	0.85
90th percentile	0	0	0	0	0.71
Programmer Capability (PCAP)					
15th percentile	0	0	0	0	1.34
35th percentile	0	0	0	0	1.15
55th percentile	1	1	1	1	1.00
75th percentile	0	0	0	0	0.88
90th percentile	0	0	0	0	0.76
Required Software Reliability (RELY)					
Slight inconvenience	0	0	0	0	1.23
Low, easily recoverable losses	0	0	0	0	1.10
Moderate, easily recoverable losses	1	1	1	1	1.00
High financial loss	0	0	0	0	0.99
Risk to human life	0	0	0	0	1.07
Documentation match to life-cycle needs(DOCU)					
Many life-cycle needs uncovered	0	0	0	0	0.81
Some life-cycle needs uncovered	0	0	0	0	0.91
Right-sized to life-cycle needs	1	1	1	1	1.00
Excessive for life-cycle needs	0	0	0	0	1.11
Very excessive for life-cycle needs	0	0	0	0	1.23
Platform Volatility(PVOL)					
Major change every 12 month; minor change every 1 month	0	0	0	0	0.87
Major: 6 month; minor: 2 week	1	1	1	1	1.00
Major: 2 month; minor: 1 week	0	0	0	0	1.15
Major: 2 week; minor: 2 days	0	0	0	0	1.30
Personnel continuity (PCON)					
48% / year	0	0	0	0	1.29
24% / year	0	0	0	0	1.12
12% / year	1	1	1	1	1.00
6% / year	0	0	0	0	0.90
3% / year	0	0	0	0	0.81

Continued on next page

Table E.2 – continued from previous page

	Task 1	Task 2	Task 3	Task 4	Multiplier
Application experience(AEXP)					
<= 2 months	0	0	0	0	1.22
6 months	0	0	0	0	1.10
1 year	1	1	1	1	1.00
3 years	0	0	0	0	0.88
6 years	0	0	0	0	0.81
Platform experience (PEXP)					
<= 2 months	0	0	0	0	1.19
6 months	0	0	0	0	1.09
1 year	1	1	1	1	1.00
3 years	0	0	0	0	0.91
6 years	0	0	0	0	0.85
Language experience (LEXP)					
<= 2 months	0	0	0	0	1.20
6 months	0	0	0	0	1.09
1 year	1	1	1	1	1.00
3 years	0	0	0	0	0.91
6 years	0	0	0	0	0.84
Use of Software Tools(TOOL)					
Edit, code, debug	0	0	0	0	1.17
Simple, frontend, backend CASE, little integration	0	0	0	0	1.09
Basic lifecycle tools moderately integrated	1	1	1	1	1.00
Strong, mature lifecycle tools, moderately integrated	0	0	0	0	0.90
Strong, mature, proactive lifecycle tools, well integrated with processes, methods, reuse	0	0	0	0	0.78
Multisite development(SITE)					
Collocation: International; Communication: Some phone, mail	0	0	0	0	1.22
Collocation: Multi-city and multi-company; Communication: Individual phone, FAX	0	0	0	0	1.09
Collocation: Multi-city or multi-company; Communication: Narrow-band email	0	0	0	0	1.00
Collocation: Same city or metro area; Communication: Wide-band electronic communication	0	0	0	0	0.93
Continued on next page					

Table E.2 – continued from previous page

	Task 1	Task 2	Task 3	Task 4	Multiplier
Collocation: Same building or complex; Communication: Wide-band electronic communication	0	0	0	0	0.86
Collocation: Fully collocated; Communication: Interactive multimedia	0	0	0	0	0.80
Sum of Multipliers (SM)	14.94	14.60	14.60	14.47	
Programmer's unfamiliarity (UNFM)					
Completely familiar	0	0	0	0	0.00
Mostly familiar	0	0	0	1	0.20
Somewhat familiar	0	0	1	0	0.40
Considerably familiar	1	0	1	0	0.60
Mostly unfamiliar	0	0	0	0	0.80
Completely unfamiliar	0	0	0	0	1.00
Software understanding (SU)					
Very low	0	0	0	0	50.00
Low	0	0	0	0	40.00
Nominal	1	1	1	1	30.00
High	0	0	0	0	20.00
Very high	0	0	0	0	10.00

F Appendix F

This appendix contains the experiment registration form that was provided on-line for the students to register and the booklet that was used during the experiment. The content of the booklet used in the experiment for describing the context, the system and the tasks were the same for both the groups. Therefore, to avoid the repetition in this appendix, we have removed the same content in the booklet of Group B (i.e., pages from 1 to 8). It starts from the walk through of HyEEASe.

Experiment registration

Experiment Date: April 30th, 2018

Experiment Place: SPPM Lecture room: Building 11 Room 207

Contact: Binish Tanveer (binish.tanveer@iese.fraunhofer.de)

Introduction:

Software effort/ time estimation is a fundamental phase for the development of any software. However, if not done properly it may cause delays in software delivery, budget overruns as well as serious implications for employees/ companies. Many software estimation methods have been proposed in academia each claiming to provide improved effort/ time estimation making them more realistic for project planning purposes. In this direction, I am conducting an experiment.

Objective:

The objective of the experiment is to compare few software estimation methods and to find which one is more useful for estimating the development effort.

Your participation will provide us with an opportunity to analyze these estimation methods. Therefore, we invite you to participate in the experiment scheduled on April 30th in the SPPM lecture class. With this experiment, you will get an opportunity to learn and gain insights of these state of the art software estimation methods. We will present you these methods on May 3rd in the SPPM exercise class.

Registration:

In order to register for the experiment, please fill out this short questionnaire. The questionnaire comprises of questions regarding your demographic information, theoretical and practical knowledge. We need this information for better analyzing the experiment results. The registration will close on April 24th.

It will take about 7-8 minutes to fill out the questionnaire. We would ask you to please answer all the questions completely and to the best of your knowledge.

Data confidentiality:

Only the researchers will have access to the information that will be collected from this questionnaire and the experiment. Personal information about you and your given answers will be kept confidential. Any information about you will have an identification number instead of your name. It will not be shared with or given to anyone except the research team.

Sharing the Results:

- Information (i.e. raw data that is collected directly from this experiment)
None of the data that you provide us will be shared with anybody outside the research team and nothing will be attributed to you by name.
- Knowledge (i.e. the aggregated results of the experiment)
This knowledge will be published as part of Binish Tanveer's research work. In addition, we will publish the anonymized results so that other interested people may learn from our research e.g. conferences, journals.

Right to refuse or withdraw:

Participation in this study is entirely voluntary. It is your choice whether to participate or not. You do not have to take part in this research if you do not wish to do so. Choosing not to participate will not affect your overall assessment in the course in any way.

However if you participate, you will get 5% bonus points in the SPPM lecture.

Participant demographic information

Contact email:

Name:

Matriculation number:

Educational background

1. Which program are you currently enrolled in?
 - ☐ Bachelor
 - ☐ Master
2. What is your major in your program e.g. BS/MS Computer science/ Informatik, Cognitive sciences etc.
Please specify: _____
3. Which semester of the program are you in? e.g. First/1st semester
Please specify: _____

Theoretical knowledge

1. Which of the following have you taken/ are you taking?
 - ☐ None
 - ☐ BS – Software development (1/2/3/4)
 - ☐ BS – SW development project
 - ☐ BS – Foundations of Software Engineering
 - ☐ BS – Seminar: Software Engineering
 - ☐ BS – Project Management
 - ☐ BS – Project - Agile Methods
 - ☐ MS – Project – Software Systems
 - ☐ MS – Process Modeling
 - ☐ MS – Empirical Model Building and Methods
 - ☐ MS – System and Software Architecture
 - ☐ MS –Seminar: Software Engineering
 - ☐ MS – Quality Management of Software and Systems
 - ☐ MS – Project: Software Engineering (Team based Software Development/ Software Evolution etc..)
 - ☐ MS – Project and Process management
 - ☐ Other
2. Which of the software development models are you familiar with?
 - ☐ None
 - ☐ Waterfall model
 - ☐ Spiral model
 - ☐ Rational Unified Process (RUP)
 - ☐ Agile development process
 - ☐ Other: _____
3. Which of the agile methods are you familiar with?

- ☐ None
- ☐ Scrum
- ☐ Test driven development (TDD)
- ☐ Extreme programming (XP)
- ☐ Other: _____

4. Are you familiar with software development effort or time estimation methods?

- ☐ No
- ☐ Yes, I have read about it
- ☐ Yes, I have used it while working on a project in a course
- ☐ Yes, I have used it while working on a project in a company
- ☐ Other: _____

5. Which of the software development effort or time estimation methods are you familiar with?

- ☐ None
- ☐ Expert based estimation (e.g., Ad hoc, Wide band Delphi, Planning poker)
- ☐ Data based algorithm model (e.g., COCOMO)
- ☐ Combination of both expert based and data based (e.g., CoBRA)
- ☐ Other: _____

6. Which of the expert based estimation methods are you familiar with?

- ☐ None
- ☐ Ad hoc expert estimation
- ☐ Wide band Delphi
- ☐ Planning poker
- ☐ Other: _____

7. In Planning Poker, do experts estimate a requirement by consulting an estimation model?

- ☐ Yes
- ☐ No
- ☐ I don't know

8. Which code metrics are you familiar with?

- ☐ None
- ☐ Lines of code (LOC)
- ☐ Cyclomatic complexity (McCabe cyclomatic complexity)
- ☐ Coupling
- ☐ Cohesion
- ☐ Other: _____

9. How did you get familiar with code metrics?

- ☐ I am not familiar with any code metrics
- ☐ I have read about them
- ☐ I have calculated one/some of them while working on a project in a course
- ☐ I have calculated one/some of them while working on a project in a company
- ☐ Other: _____

Practical knowledge

1. How much experience do you have in software development?
 - ☐ None
 - ☐ Less than 1 year
 - ☐ 1-3 years
 - ☐ 3-5 years
 - ☐ More than 5 years
 - ☐ Other: _____

2. How much experience do you have in agile software development?
 - ☐ None
 - ☐ Less than 1 year
 - ☐ 1-3 years
 - ☐ 3-5 years
 - ☐ More than 5 years
 - ☐ Other: _____

3. Which of the agile methods have you been working with?
 - ☐ None
 - ☐ Scrum
 - ☐ Test driven development (TDD)
 - ☐ Extreme programming (XP)
 - ☐ Other: _____

4. In what role/s have you been working?
 - ☐ None
 - ☐ Developer
 - ☐ Tester
 - ☐ Architect/ Designer
 - ☐ Project Manager
 - ☐ Requirement engineer
 - ☐ Scrum master
 - ☐ Product owner
 - ☐ Other: _____

5. Have you ever estimated software development effort or time in practice?
 - ☐ No
 - ☐ Yes, I have estimated development effort or time while working on a project in a course
 - ☐ Yes, I have estimated development effort or time while working on an Open Source Software project
 - ☐ Yes, I have estimated development effort or time while working on a project in a company
 - ☐ Other: _____

6. How often did you estimate software development effort or time?
- ☐ Never
 - ☐ Rarely
 - ☐ Frequently
7. Which estimation methods have you been using from the following categories?
- ☐ None
 - ☐ Expert based estimation (e.g., Ad hoc, Wide band Delphi, Planning poker)
 - ☐ Data based algorithm model (e.g., COCOMO)
 - ☐ Combination of both expert based and data based (e.g., CoBRA)
 - ☐ Other: _____
8. Please specify the name of the estimation method/s that have you been using.
- _____
9. What have you been estimating?
- ☐ None
 - ☐ Size (code) of the project/ change request
 - ☐ Complexity of the code
 - ☐ Total development effort
 - ☐ Total development time
 - ☐ Total development cost
 - ☐ Other: _____
10. Have you used any tool to make estimations?
- ☐ No
 - ☐ Yes

Name:

Matriculation no:

Group A

Experiment in SPPM course

Date: April, 30th 2018

Booklet - Group A and Group B

Name:

Matriculation no:

Group A

Thank you for participating in the experiment.

Experiment objective:

The objective of the experiment is to compare software estimation methods and to find which one is more useful for estimating the development effort. Your participation will provide us with an opportunity to analyze these estimation methods. During this experiment, you will get an opportunity to learn and gain insights of these state of the art software estimation methods. We will further discuss these methods with you on May 3rd in the SPPM exercise class. Please note that your personal information and your answers will be kept confidential. Moreover, experiment results will not affect your assessment in the course.

Before starting the experiment, we will shortly describe software development effort estimation.

Software development effort estimation:

It is the process of predicting the most realistic amount of effort required to develop or maintain software based on incomplete and/or uncertain requirements/features. The effort can be estimated, e.g., in person-hours, story points (SP), person hours (PH) or ideal days etc.

Software development effort estimates are used for project bidding, budgeting and planning. These are critical practices in the software industry because poor budgeting and planning often have dramatic consequences. When budgets and plans are too pessimistic, business opportunities can be lost, while over-optimism may lead to significant losses [1]. Three kinds of estimation methods exist:

- **Expert-based** (e.g. Adhoc, Wideband Delphi, Planning poker etc.) these methods rely on an expert's knowledge of the system and experience of estimation in the past and their subjective judgment.
- **Data-based models** (e.g. COCOMO etc.) these methods rely on mathematical models that identify and exploit relations and patterns in the historical data about estimation.
- **Hybrid** (e.g. CoBRA™[2] etc.) these methods combine expert-based and data-based methods.

In the following section, we will introduce the system and associated features that are the objects of experiment.

Name:

Matriculation no:

Group A

The system

Moodle [3]: stands for “modular object-oriented dynamic learning environment”. It is an open source, online learning management system like OLAT. It is developed on the open-source LAMP framework consisting of Linux (operating system), Apache (web server), MySQL (database), and PHP (programming language). Due to the portability of these components and the modularity of Moodle itself, it can support a wide range of operating systems, database systems, and web servers[4].

It is designed to create opportunities for rich interaction between teachers and students. Its goal is to give teachers and students the tools they need to teach and learn, respectively.

Moodle [5] is explained below by using an analogy with Lego bricks.





- Imagine Moodle as a platform that comes with a huge set of Lego bricks.
- On this platform, we can have different foundations (called courses). This is where we put our Lego bricks on.



Let us imagine we can do four basic things with four colored Lego bricks.

Brick type	Examples:
 Store	We can store...  Files,  folders,  database,  webpages etc.
 Communicate	We can communicate through...  RSS,  forums,  calendar,  messaging etc.
 Collaborate	We can collaborate through...  Blog,  wiki,  workshop,  glossary etc.
 Evaluate	We can evaluate with...  Grade,  quiz,  assignment,  survey etc.

- We can have as many bricks we want. And, we can get many other compatible bricks too e.g.

 polls,  individual learning plan,  open meetings,  timers etc.

- We can arrange the bricks that fit in any way that suits our educational purpose.
- We decide who gets in the system and what they can do, therefore we have roles e.g.

- An admin can do, see edit anything on the system.
- A teacher can do, see and edit anything in the course.
- A student can do, see and edit anything in the course as assigned by the teacher.
- A guest can only look at few parts, permissions for a parent or any other role can be specified



Name:

Matriculation no:

Group A

Moodle architecture [4] : Moodle follows a basic three-tier architecture as shown in Figure 1 below. It is structured as an application core, surrounded by numerous plugins to provide specific functionality. Moodle is designed to be highly extensible and customizable without modifying the core libraries but through the plugin architecture.

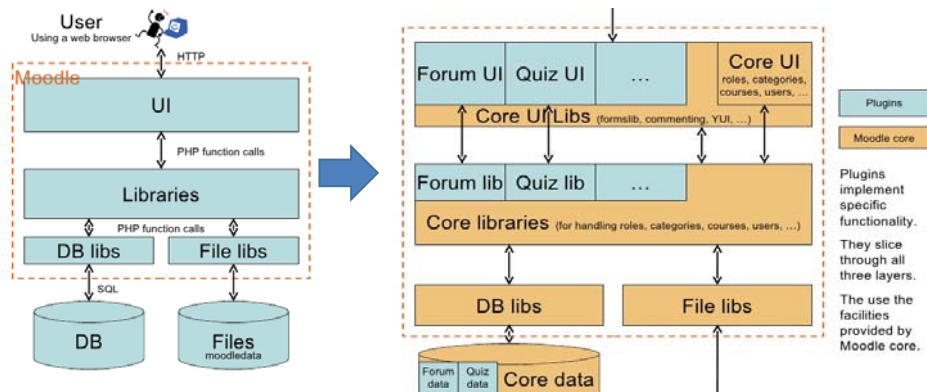


Figure 1: Moodle architecture³

Plugins in Moodle are of specific types. For example, an authentication plugin and an activity module will communicate with Moodle core using different APIs, tailored to the type of functionality the plugin provides. Functionality common to all plugins (installation, upgrade, permissions, configuration etc.) are, however, handled consistently across all plugin types. Following are some types of plugins:

- Activity modules
- Assignment types
- Course format
- Authentication plugins
- Gradebook
- Plagiarism detection plugins
- Quiz reports
- Workshop forms

Moodle core provides all the infrastructure necessary to build a learning management system. It implements the key concepts that all the different plugins will need to work with. These include but not limited to: Courses and activities, users (e.g. teacher, student), course enrolment, user functionality etc.

In the next section, we will describe the feature in focus today for this experiment.

Name:

Matriculation no:

Group A

Feature in focus today: Plagiarism prevention [6]

***Plagiarism** is the act of taking someone else's work or ideas and passing them off as your own. These days some students plagiarize material available on the internet. They may submit content that they have copied without acknowledging the real author of the work. Plagiarism check detects when this form of cheating or academic dishonesty has happened.*

In Moodle, plagiarism is detected through “plagiarism prevention plugin” with the following **basic functionality**.

- **Plagiarism prevention plugin:**
 - Currently the plugin only supports the assignment activity.
 - Currently the plugin is only able to detect whether the text in an assignment is copied from a source on the Internet.

Required enhanced features:

1. **Feature 1:** Extend basic functionality i.e. in addition to detecting the copied text, also generate:
 - a. Report (with statistics) indicating what parts of assignment have been copied.
 - b. Analyze the plagiarized part and suggest the action category (acceptable, acceptable after revision, not acceptable).
 - c. Analyze the plagiarized part and visualize effected parts with percentages (graphs).
2. **Feature 2:** Extend the feature 1 to support quizzes (individual) and/or workshop (group work) activity submission.

In the next section, we will describe the context and associated tasks.

Name:

Matriculation no:

Group A

Your profile:

1. Assume you work for the company TUKLSOFT.
2. You have joined TUKLSOFT as a developer.
3. You are provided only with the previous description of the system and feature 1 to be implemented.

Context:**The company:**

1. TUKLSOFT is a software development company.
2. It is extending the above mentioned large-scale Moodle system.
3. TUKLSOFT is following agile development approach using Scrum. The development takes place in a three-week periods called sprint.

Sprint planning and estimation

1. Every sprint, the team takes a number of the most urgent user stories/ features from the backlog (user stories/ features document) to work on during that sprint.
2. At the start of a sprint, there is a period of planning and estimation.
 - a. The team estimates the effort/ size or complexity of the features.
 - b. Starting with small features and incrementally adding larger ones.
 - c. All activities are taken into account (e.g. development, test, integration, documentation etc.)
3. Single estimates are done using, e.g., “planning poker” i.e. every member plays a card containing a value similar to values in Fibonacci series (0, 1, 2, 3, 5, 8...). These values represent effort estimation units like story points (SP), person hours (PH) or ideal days etc. (A detailed explanation of planning poker will be provided to you later).
4. During the sprint, the team meets daily to discuss solutions and progress, as well as to organize testing and peer reviews of code.
5. The team has a Scrum master to help everyone stay organized and to protect the team from distracting influences during the sprint.

Your job:

You are assigned the task to review the estimated effort for feature 1 made by the TUKLSOFT’s scrum team in the sprint planning and estimation.

Note:

- The motive behind your job is to let you gain knowledge and experience of how estimation is done at TUKLSOFT.
- With this experience, you will be able to analyze the method and its output i.e. the estimates.
- Please provide all the information asked in the experiment to the best of your knowledge and based on your experience you gather during this experiment.
- Please note that there is no right or wrong answer and therefore, your performance will not be judged in any way.
- Please do not leave any question unanswered.
- Please write clearly.

In the next section, we will describe the tasks that you have to perform.

Name:

Matriculation no:

Group A

- 1. Please read the provided description of the system (i.e. Moodle) and feature 1. Take few minutes to understand both Moodle and feature 1. Please contact the experimenter if you have any question regarding Moodle or feature 1.**
- 2. As soon you have understood Moodle and feature 1, please raise your hand so we can provide you the tasks you are requested to performed.**

Group A

Task 1

Start time: _____ [HH:MM]

Please answer the following questions.

1. Based on the provided description of Moodle and feature 1: What information will you consider for estimating the effort required for implementing feature 1?
2. What other information would you like to have for estimating the effort required for implementing feature 1?

Task 1

End time: _____ [HH:MM]

**Now please raise your hand so we can provide you with the next task.
Please handover this page to the experimenter.**

Name:

Matriculation no:

Group A

Walkthrough of Planning poker for Group A

Name:

Matriculation no:

Group A

Please specify the start and end time for the following task.

Task 2

Start time: _____ [HH:MM]

Now we will provide a walkthrough of the planning poker method for estimating feature 1 and ask a few questions about it.

Planning poker [7][8] is an agile estimating and planning technique that is consensus-based. It guarantees that all members actively participate through playing a game.

Setting:

- Before, the game starts, a set of cards is given to each member of the team.
- Each set of cards contain a distinct number (similar to Fibonacci sequence) i.e. 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100.
- These numbers represent a relative estimation of the story/feature. Zero means the story/feature is a minor fix; twenty means that the story/feature needs to be broken into smaller ones.
- The rationale behind using Fibonacci is the intent to create uncertainty, which will encourage discussion among team members, especially with large numbers.
- These numbers can have any unit i.e. story points (SP), person hours (PH), ideal days, or other units in which the team estimates the effort required to implement a story/feature.

Planning poker walkthrough for estimating feature 1 during sprint planning (shown in Figure 1).

The sprint planning meeting is currently on going at TUKLSOFT following the planning poker setting described above. The team comprises four developers (Tim, Jenny, Thomas and Sara). Thomas is the scrum master.

1. In step 1, see Figure 1, the planning poker game starts when the product owner or scrum master reads a feature/ story to the estimators in the team. **In case of TUKLSOFT, Thomas read the feature 1 to the estimators.**
2. Each estimator has a deck of planning poker cards.
3. In step 2, see Figure 1, the estimators discussed the feature 1, and asked Thomas some follow-up questions regarding the feature. Furthermore, Tim who has previously worked on the “plagiarism plugin” was asked his opinion about implementing feature 1. Tim said, *“I remember working on this component in the past, the code and the underlying logic is pretty complex. It took me some time to understand it since the documentation did not help much. Additionally, I think this plugin requires functionality like authentication from other plugins so we should keep that in mind while working with it again for feature 1. It may require major changes.”* Jenny asked Thomas, *“What is the priority? Do we need to provide feature 1-part b/ c in this sprint as well or can they be deferred?”* Thomas replied that the complete feature 1 is urgently required. The discussion continued for some more time.
4. In step 3, see Figure 1, when the feature 1 was sufficiently discussed, each estimator privately selected one card to represent their estimate.
5. All cards were then revealed at the same time.
6. Generally, if all estimators select the same value, that becomes the effort estimate for a certain feature. If not, the estimators discuss their estimates. The high and low estimators should especially share their reasons. **In case of feature 1, Jenny and Sara showed 3 SP card where Thomas showed 5 SP card and Tim showed 8 SP card.** They discussed the rationales of their different estimates. Jenny and Sara said *“we don’t know much about plagiarism plugin, yet, in feature 1-part c, it is asked to add the visualizations, that means we need to call visualizations component in addition. The visualization component is not so complex itself, so we estimated feature 1 as 3 SP.”* Thomas said *“In order to implement the feature 1 completely in this sprint, I*

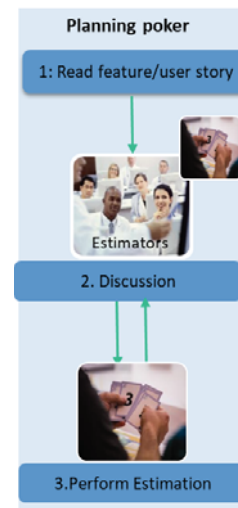


Figure 1: Planning poker

Name:

Matriculation no:

Group A

would say 5 SP as we all need to make sure all implementation in the component is done smoothly, as Tim is fully occupied with other features in this sprint". After this discussion, the estimators ran another round of cards selection.

7. Each estimator reselected an estimate card, and all cards were again revealed at the same time.
8. Generally, the process (steps 1-8) is repeated until consensus is achieved or until the estimators decide that estimating and planning of a particular feature needs to be deferred until additional information can be acquired. **In case of feature 1, the estimators settled on 8 SP estimate and so it became the final estimate.**

As soon as you have understood the walkthrough of planning poker, write the end time and please raise your hand so we can provide you with the tasks to be performed.

Task 2

End time: _____ [HH:MM]

Name:

Matriculation no:

Group A

Please specify the start and end time for the following task.

Task 3

Start time: _____ [HH:MM]

Please answer the following multiple choice questions:

The goal of planning poker is

- ☐ To get each team member's input on how difficult the story is.
- ☐ To estimate the complexity of a user story by group consensus.
- ☐ To estimate user stories/features in story points.
- ☐ All of the above
- ☐ None of the above

The result of the planning poker is...

- ☐ Development of a common understanding of the team about the user stories/features.
- ☐ Development of a sprint plan.
- ☐ Complexity estimates of user stories/features.
- ☐ All of the above
- ☐ None of the above

Task 3

End time: _____ [HH:MM]

Now please raise your hand so we can provide you with the next task.

Please handover this page to the experimenter.

Name:

Matriculation no:

Group A

Please specify the start and end time for the following task.

Task 4

Start time: _____ [HH:MM]

Assume you are a part of the development team and you are reviewing the estimate provided by the team for feature 1 i.e. 8 SP.

Please answer the following:

1. Do you agree with the provided estimate for feature 1 (i.e. 8 SP)?

- ☐ Yes
- ☐ No
- ☐ I don't know

2. What information (not yet considered by the team) would you consider while estimating feature 1? (multiple choice)

- ☐ None
- ☐ Developer's development experience
- ☐ Developer's estimation experience
- ☐ Developer's knowledge of the system (Moodle)
- ☐ Complexity of feature 1 and the system (Moodle)
- ☐ No. of potentially impacted components
- ☐ Size of the potentially impacted components
- ☐ Dependencies among the potentially impacted components
- ☐ Complexity of the potentially impacted components
- ☐ Effort estimates made in the past for the same impacted components
- ☐ Others, please specify:

3. If your answer to the previous question was not "none", then please answer:

After considering the additional information listed in the previous question, I would expect the estimate for feature 1 to be:

- ☐ Significantly greater than 8 story points
- ☐ Greater than 8 story points
- ☐ Equal to 8 story points
- ☐ Less than 8 story points
- ☐ Significantly less than 8 story points

Now assume that the team also estimated effort for feature 2. Please read the provided description of Moodle and feature 2. **Take few minutes to understand feature 2.**

Name:

Matriculation no:

Group A

The team estimated the effort for feature 2 to be 8 SP.

Please compare the provided estimate for feature 2 (i.e. 8 SP) with feature 1 (i.e. 8 SP) and answer the following questions:

1. Do you agree with the estimate for feature 2?

- ☐ Yes
- ☐ No, the estimate of feature 2 should be greater than the estimate for feature 1
- ☐ No, the estimate of feature 2 should be less than the estimate for feature 1
- ☐ I don't know

2. Please provide the rationale for your answer to the previous question.

Task 4

End time: _____ [HH:MM]

**Now please raise your hand so we can provide you with the next task.
Please handover this page to the experimenter.**

Name:

Matriculation no:

Group A

Please specify the start and end time for the following task.

Task 5

Start time: _____ [HH:MM]

To what degree you agree or disagree with the following statements:

I think the planning poker method...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... is clear	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is understandable	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... requires a lot of mental effort to apply it	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is easy to use	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I think the planning poker method...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... has an appropriate level of detail	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is missing critical data/information that would be very useful to estimate efforts of enhanced features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In general, complete for estimating effort of enhanced features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

After knowing planning poker method...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
I am convinced I can learn over time, which factors influence estimation of enhanced features.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am convinced it will help me learn fast about the factors influencing estimation of enhanced features.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Task 5

End time: _____ [HH:MM]

Thank you very much for your participation!

We will present the design of this experiment on May 3rd in the exercise class,

building 13-222, 13:45 – 15:15.

References

- [1] Jorgensen, Magne. "What we do and don't know about software development effort estimation." *IEEE software* 31.2 (2014): 37-40.
- [2] CoBRA™: <https://cobra.fraunhofer.de/method/index.html>
- [3] Moodle: https://docs.moodle.org/dev/Moodle_architecture
- [4] Moodle architecture: https://www.packtpub.com/mapt/book/hardware_and_creative/9781847195623/3/ch03/v1sec01/moodle-architecture
- [5] Moodle explained: https://www.slideshare.net/moodiefan/what-is-moodle-explained-with-lego-presentation?next_slideshow=1
- [6] Plagiarism: https://docs.moodle.org/34/en/Plagiarism_prevention_FAQ
- [7] Planning poker: Cohn, Mike. *Agile estimating and planning*. Pearson Education, 2005.
- [8] Planning poker: <https://www.mountaingoatsoftware.com/agile/planning-poker>

Name:

Matriculation no:

Group B

Walkthrough of HyEEASe for Group B

Name:

Matriculation no:

Group B

Please specify the start and end time for the following task.

Task 2

Start time: _____ [HH:MM]

Now we will provide a walkthrough of HyEEASE for estimating feature 1 and ask a few questions about it.

HyEEASE [7] (hybrid effort estimation in agile software development) is a tool based hybrid estimation method created to support planning poker. It is hybrid as it combines expert knowledge and code-based change impact analysis. Change impact analysis provides information of the parts/components of the system that are “likely to be” (or potentially be) impacted by a given change request i.e. a defect/enhanced feature.

HyEEASE tool performs change impact analysis and provides this information to the estimators during planning poker as shown in the Figure 2. We have explained this below by starting with planning poker-

Planning poker [8][9] is an agile estimating and planning technique that is consensus-based. It guarantees that all members actively participate through playing a game.

Setting:

- Before, the game starts, a set of cards is given to each member of the team.
- Each set of cards contain a distinct number (similar to Fibonacci sequence) i.e. 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100.
- These numbers represent a relative estimation of the story/feature. Zero means the story/feature is a minor fix; twenty means that the story/feature needs to be broken into smaller ones.
- The rationale behind using Fibonacci is the intent to create uncertainty, which will encourage discussion among team members, especially with large numbers.
- These numbers can have any unit i.e. story points (SP), person hours (PH), ideal days, or other units in which the team estimates the effort required to implement a story/feature.

HyEEASE for estimating feature 1 during sprint planning (shown in Figure 1)

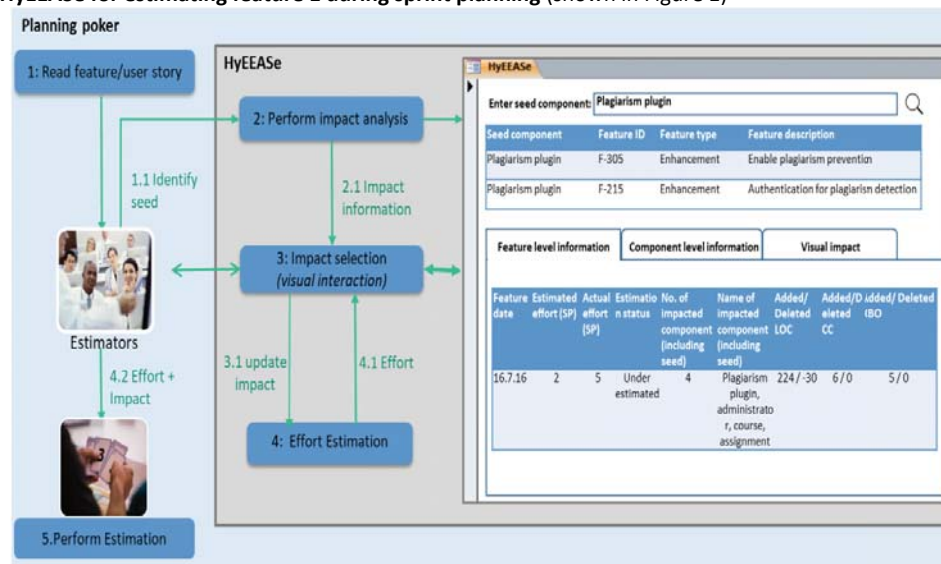


Figure 1: HyEEASE

The sprint planning meeting is currently on going at TUKLSOFT following the planning poker setting described above. The team comprises four developers (Tim, Jenny, Thomas and Sara). Thomas is the scrum master.

10

Name:

Matriculation no:

Group B

- a. In step 1, (see Figure 1), the planning poker starts when the product owner or scrum master reads a feature/ story to the estimators in the team. **In case of TUKLSOFT sprint planning, Thomas read the feature 1 to the estimators.**
- b. Each estimator has a deck of planning poker cards.
- c. The estimators discussed the feature 1, and asked Thomas some follow-up questions regarding the feature. Furthermore, Tim who has previously worked on the “plagiarism plugin” was asked his opinion about implementing feature 1. Tim said, *“I remember working on this component in the past, the code and the underlying logic is pretty complex. It took me some time to understand it since the documentation did not help much. Additionally, I think this plugin requires certain functionality from other plugins, so we should keep that in mind while working with it again for feature 1. It may require major changes. Let’s see what HyEEASe says about my thoughts.”* Jenny asked Thomas, *“What is the priority? Do we need to provide feature 1-part b/ c in this sprint as well or can they be deferred?”* Thomas replied that the complete feature 1 is urgently required. The estimators then consulted HyEEASe to get additional information.

HyEEASe workflow as shown in Figure 1 and is described below:

- In step 1.1, (see Figure 1); the estimators identify an initial “seed” which is a component (or components) that is likely to be impacted (or potentially impacted) when a given feature is to be implemented. **In this case, the given feature is feature 1 and the four estimators identified the seed as “plagiarism plugin”.** The seed was given as input to HyEEASe in the “Enter seed component” field in **Error! Reference source not found.**
- In step 2, (see Figure 1), HyEEASe performed impact analysis on Moodle code using the “plagiarism plugin” as input.
- In step 2.1, the returned impact is displayed by HyEEASe tool to the estimators. This impact comprised of information based on the given “seed” i.e. the “plagiarism plugin”. The impact information contains all the corresponding previously implemented feature/s where the “plagiarism plugin” was impacted (see “Features where seed was impacted” in **Error! Reference source not found.** **In this case, of feature 1, when “plagiarism plugin” was given as seed to HyEEASe, it found two previously implemented features i.e. “F-305” and “F-215”.** F-305 feature was about enabling plagiarism prevention and F-215 was about authentication for plagiarism plugin.
- Upon selecting the feature F-305, HyEEASe displayed the corresponding **feature level impact information** (see “Feature level information” tab in **Error! Reference source not found.**) that comprised of “Feature attributes” for F-305 with values as seen in “Attribute values”.

Name:

Matriculation no:

Group B

HyEEASe

Enter seed component:

Seed component	Feature ID	Feature type	Feature description
Plagiarism plugin	F-305	Enhancement	Enable plagiarism prevention
Plagiarism plugin	F-215	Enhancement	Authentication for plagiarism detection

Feature level information | Component level information | Visual impact

Feature date	Estimated effort (SP)	Actual effort (SP)	Estimation status	No. of impacted component (including seed)	Name of impacted component (including seed)	Added/ Deleted LOC	Added/ Deleted CC	Added/ Deleted CBO
16.7.16	2	5	Under estimated	4	Plagiarism plugin, administrator, course, assignment	224 / -30	6 / 0	5 / 0

Callouts:

- Features where seed was impacted: Points to the 'Actual effort (SP)' value of 5.
- Feature attributes: Points to the 'No. of impacted component (including seed)' value of 4.
- Attribute values: Points to the 'Added/ Deleted LOC' value of 224 / -30.

Figure 2: HyEEASe feature level information

As seen in **Error! Reference source not found.**, “Feature level information” tab, HyEEASe informed the estimators that F-305 was estimated with 2 SP but actually took 5 SP and so was underestimated. This implementation impacted four components in Moodle code i.e. “Plagiarism plugin” (seed), administrator, course, assignment”. Due to this implementation, impact factors were also significantly changed across all of the above mentioned four impacted components in Moodle code i.e.:

- Altogether 224 lines of code (LOC) were added, 30 were deleted
- Cyclomatic complexity (CC) was also increased by 6 units and coupling between objects (CBO) was increased by 5 units

- HyEEASe also displayed **component level impact information** (see **Error! Reference source not found.**, “Component level information” tab) that comprised of “Component attributes” and their values in “Attribute values” for “plagiarism plugin”:

Name:

Matriculation no:

Group B

The screenshot shows the HyEEASe interface. At the top, there is a search bar labeled 'Enter seed component:' with 'Plagiarism plugin' entered. Below this is a table with the following data:

Seed component	Feature ID	Feature type	Feature description
Plagiarism plugin	F-305	Enhancement	Enable plagiarism prevention
Plagiarism plugin	F-215	Enhancement	Authentication for plagiarism detection

Below this table are three tabs: 'Feature level information', 'Component level information' (which is selected), and 'Visual impact'. The 'Component level information' tab displays a table with the following data:

Seed component	Feature ID	Individual effort	Added/ Deleted LOC	Added/Deleted CC	Added/ Deleted CBO
Plagiarism plugin	F-305	2	50 / -15	4 / 0	5 / 0

Annotations with arrows point to specific parts of the interface:

- 'Seed and corresponding feature' points to the 'Plagiarism plugin' row in the first table.
- 'Component attributes' points to the 'Individual effort' column in the second table.
- 'Attribute values' points to the '4 / 0' value in the 'Added/Deleted CC' column.

Figure 3: HyEEASe component level information

As seen in **Error! Reference source not found.**, “Component level information” tab, HyEEASe informed the estimators that due to F-305, when “plagiarism plugin” was changed, it individually took 2 SP effort out of the 5 SP total effort of the F-305. This implementation also changed the impact factors significantly in the “plagiarism plugin” in the Moodle code i.e.:

- Altogether 50 lines of code (LOC) were added, 15 were deleted.
- Cyclomatic complexity (CC) was increased by 4 units and coupling between objects (CBO) was increased by 5.

Name:

Matriculation no:

Group B

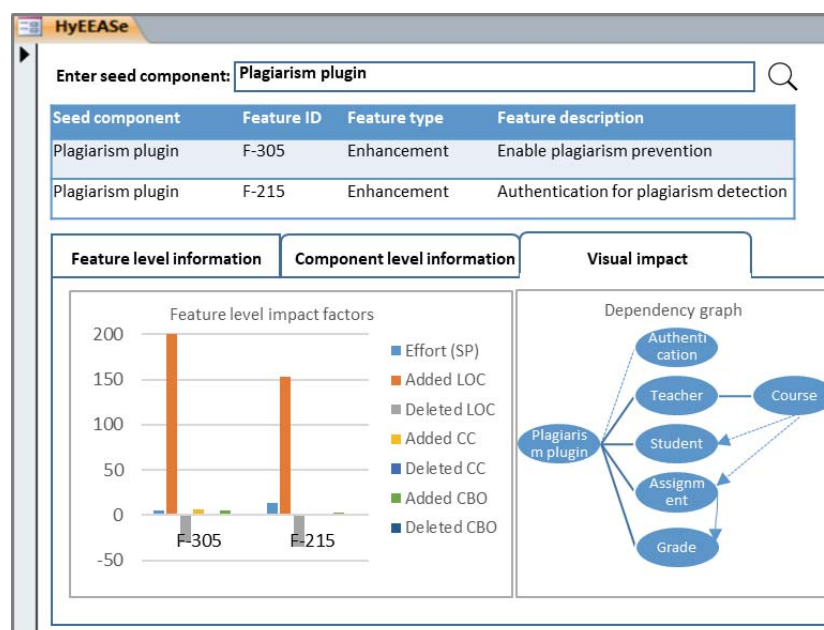


Figure 4: HyEEASE view impact tab

- HyEEASE also visualizes the impact information through a variety of charts (bar/ column/ pivot charts), (see “Visual impact” tab “Feature level impact factors” (left) in **Error! Reference source not found.**). With this information in one holistic view, the estimators analyzed all the changed impact factors for all previously implemented features (i.e. F-305 and F-215) where “plagiarism plugin” was one of the impacted components among others. **In case of F-305, the chart shows that actual effort was 5 SP, added LOC were 224, deleted LOC were 30, added CC was 6 and added CBO was 5 across all four impacted components (Plagiarism plugin, administrator, course, assignment).**
- Additionally, HyEEASE displays dependency graphs, showing the code internal structure, of impacted seed component (see in **Error! Reference source not found.**, “Visual impact” tab “Dependency graph” (right). **In this case, Tim asked to explore the “plagiarism plugin” through dependency graphs. From where, they found out that, components like “authentication, teacher, student, assignment and grade” were tightly coupled with “plagiarism plugin”.**
- In step 3, (see Figure 1), the estimators, while discussing, can select/deselect which of the previously impacted components may potentially be impacted again while implementing the feature 1.
- In step 3.1 and step 4, based on estimator’s selection is step 3, HyEEASE re-performs the impact analysis, updates the charts and dependency graphs, calculates/recalculates the individual effort of the associated components and displays to the estimators.
- d. In step 5, (see Figure 1) when the feature 1 was sufficiently discussed under the light of information provided by HyEEASE, each estimator privately selected one card to represent their estimate.
- e. All cards were then revealed at the same time.
- f. Generally, if all estimators select the same value, that becomes the effort estimate for a certain feature. If not, the estimators re-consult HyEEASE impact information and then discuss their estimates. The high and low estimators should especially share their reasons. **In case of feature 1, Jenny and Sara showed 3 SP card where Thomas showed 5 SP card and Tim showed 8 SP card.**

Name:

Matriculation no:

Group B

They discussed the rationales of their different estimates. Jenny and Sara said, *"We did not work with plagiarism plugin before, however, in feature 1-part c, it is asked to add the visualizations that means we need to call visualizations component, which is not so complex itself. After reading the description of feature 1 we assessed it smaller than F-305 therefore, we estimated it as 3 SP."* Thomas said, *"HyEEASe showed us that 3 other components were also impacted when 'plagiarism plugin' was impacted due to F-305. In order to implement the feature 1 completely, in this sprint, I would say 5 SP as we all need to make sure all implementation in the component is done smoothly as Tim is fully occupied with other features in this sprint". Tim said "and the column chart of the impact factors of the previous features are suggesting that perhaps for feature 1 we might be adding a lot more complexity to the existing plagiarism plugin".*

- g. After further discussion, each estimator reselected an estimate card, and all cards were again revealed at the same time.
- h. Generally, the process (steps a-h) is repeated until consensus is achieved or until the estimators decide that estimating and planning of a particular feature needs to be deferred until additional information can be acquired. **In case of feature 1, the estimators finally settled on 8 SP estimate and so it became the final estimate.**

As soon as you have understood the walkthrough of HyEEASe, write the end time and please raise your hand so we can provide you with the tasks to be performed.

Task 2

End time: _____ [HH:MM]

Name:

Matriculation no:

Group B

Please specify the start and end time for the following task.

Task 3**Start time:** _____ [HH:MM]

Please answer the following multiple choice questions:

The goal of planning poker is...

- ☐ To get each team member's input on how difficult the story is.
- ☐ To estimate the complexity of a user story by group consensus.
- ☐ To estimate user stories/features in story points.
- ☐ All of the above
- ☐ None of the above

The result of the planning poker is...

- ☐ Development of a common understanding of the team about the user stories/features.
- ☐ Development of a sprint plan.
- ☐ Complexity estimates of user stories/features.
- ☐ All of the above
- ☐ None of the above

HyEEASE provides the team with following information:

- ☐ Impact information of the previously implemented features.
- ☐ Identification of the "seed" component.
- ☐ Effort of the previously implemented features.
- ☐ Component level effort information
- ☐ Estimated effort of the features to be implemented
- ☐ All of the above
- ☐ None of the above

Task 3**End time:** _____ [HH:MM]

**Now please raise your hand so we can provide you with the next task.
Please handover this page to the experimenter.**

Name:

Matriculation no:

Group B

Please specify the start and end time for the following task.

Task 4**Start time:** _____ [HH:MM]

Assume you are a part of the development team and you are reviewing the estimate provided by the team for feature 1 i.e. 8 SP

Please answer the following:

1. Do you agree with the provided estimate for feature 1 (i.e. 8 SP)?

- ☐ **Yes**
- ☐ **No**
- ☐ **I don't know**

2. What information (not yet considered by the team) would you consider while estimating feature 1? (multiple choice)

- ☐ None
- ☐ Developer's development experience
- ☐ Developer's estimation experience
- ☐ Developer's knowledge of the system (Moodle)
- ☐ Complexity of feature 1 and the system (Moodle)
- ☐ No. of potentially impacted components
- ☐ Size of the potentially impacted components
- ☐ Dependencies among the potentially impacted components
- ☐ Complexity of the potentially impacted components
- ☐ Effort estimates made in the past for the same impacted components
- ☐ Others, please specify:

3. If your answer to the previous question was not "none", then please answer:

After considering the additional information listed in the previous question, I would expect the estimate for feature 1 to be:

- ☐ Significantly greater than 8 story points
- ☐ Greater than 8 story points
- ☐ Equal to 8 story points
- ☐ Less than 8 story points
- ☐ Significantly less than 8 story points

Now assume that the team also estimated effort for feature 2. Please read the provided description of Moodle and feature 2. **Take few minutes to understand feature 2.**

Name:

Matriculation no:

Group B

The team estimated the effort for feature 2 to be 8 SP after consulting HyEEASe. A subset of the information provided by HyEEASe is as follows:

- Seed component: Quiz
- Previously implemented feature: F-276 (Generate statistics for quiz data)
- Names of impacted components: Course, teacher
- Component level impact factors:
 - Source lines of code (Added LOC / Deleted LOC) = 125/-10
 - Cyclomatic complexity (Added CC / Deleted CC) = 3/1
 - Coupling between objects (Added CBO / Deleted CBO) = 3/0

Now please compare the provided estimate for feature 2 (i.e. 8 SP) with feature 1 (i.e. 8 SP) and answer the following questions:

1. Do you agree with the estimate for feature 2?

- Yes
- No, the estimate of feature 2 should be greater than the estimate for feature 1
- No, the estimate of feature 2 should be less than the estimate for feature 1
- I don't know

2. Please provide rationale for your answer of the previous question

Task 4

End time: _____ [HH:MM]

Now please raise your hand so we can provide you with the next task.
Please handover this page to the experimenter.

Name:

Matriculation no:

Group B

Please specify the start and end time for the following task.

Task 5

Start time: _____ [HH:MM]

To what degree you agree or disagree with the following statements:

I think HyEEASe method...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... is <i>clear</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is <i>understandable</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... requires a <i>lot of mental effort</i> to apply it	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is <i>easy to use</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I think HyEEASe method...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... has an <i>appropriate level of detail</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... is <i>missing critical data/information</i> that would be very useful to estimate efforts of enhanced features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>In general</i> , <i>complete</i> for estimating effort of enhanced features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

I think the information provided by HyEEASe is ...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
... <i>informative</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... <i>valuable</i>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<i>In general</i> , <i>useful</i> for estimating effort for enhanced features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...is <i>relevant</i> for estimating effort for enhanced features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...is <i>appropriate</i> for estimating effort for enhanced features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
...is <i>applicable</i> for estimating effort for enhanced features	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Name:

Matriculation no:

Group B

To what degree you agree or disagree with the following statements:

After knowing HyEEASE...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
I <i>learnt</i> about the factors influencing estimation of enhanced features.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am convinced I can <i>learn over time</i> , which factors influence estimation of enhanced features.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am convinced it will <i>help me learn fast about</i> the factors influencing estimation of enhanced features.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

After knowing HyEEASE ...	1: strongly disagree	2: disagree	3: neutral	4: agree	5: strongly agree	I don't know
I expect that using it will be <i>enjoyable</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I expect that the process of using it will be <i>pleasant</i> .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I expect that I will have <i>fun</i> using it.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Task 5

End time: _____ [HH:MM]

Thank you very much for your participation!

We will present the design of this experiment on May 3rd in exercise class, building 13-222,

13:45 – 15:15.

References

- [1] Jorgensen, Magne. "What we do and don't know about software development effort estimation." *IEEE software* 31.2 (2014): 37-40.
- [2] CoBRA™: <https://cobra.fraunhofer.de/method/index.html>
- [3] Moodle: https://docs.moodle.org/dev/Moodle_architecture
- [4] Moodle architecture: https://www.packtpub.com/mapt/book/hardware_and_creative/9781847195623/3/ch03lv1sec01/moodle-architecture
- [5] Moodle explained: https://www.slideshare.net/moodlefan/what-is-moodle-explained-with-lego-presentation?next_slideshow=1
- [6] Plagiarism: https://docs.moodle.org/34/en/Plagiarism_prevention_FAQ
- [7] Tanveer, Binish, Vollmer, Anna Maria, and Braun, Stefan. "A hybrid methodology for effort estimation in agile development- an industrial evaluation." *Software and System Processes (ICSSP)*, 2018 IEEE/ACM International Conference on. IEEE, 2018.
- [8] Planning poker: Cohn, Mike. *Agile estimating and planning*. Pearson Education, 2005.
- [9] Planning poker: <https://www.mountaingoatsoftware.com/agile/planning-poker>

Curriculum Vitae

Education:

European Master in Software Engineering, Technical University Kaiserslautern, Germany and Blekinge Institute of Technology (BTH), Sweden [2007-2009]

Bachelor of Science in Computer Science, Muhammad Ali Jinnah University, Pakistan [1998-2002]

Professional experience:

- Assoc. Senior Lecturer at BTH, Sweden [Sep 2019 – to date]
- Research scientist at Fraunhofer IESE, Kaiserslautern, Germany [Nov 2009- Jun 2018]
- Senior Software Engineer at Elixir Technologies, Islamabad, Pakistan [Jan 2002- Sep 2007]

Achievements:

- As Erasmus Mundus scholar received a scholarship for a joint study program titled European Masters on Software Engineering [2007-2009]. Only two candidates got selected that year from Pakistan.
- Silver Medalist (Summa cum laude) in BSc (Computer Science) on obtaining 3.68 CGPA on a 4.0 GPA scale. Scored 4.0 GPA on the scale of 4.0, four times consecutively during BS and was placed in University's Chancellor List.

Skills:

Process management: Fraunhofer IESE tools (Spearmint), Eclipse Process Framework, and Stages by Methodpark.

Statistics: RapidMiner, Knime, SPSS.

Programming: C/C++, VC++, JAVA, Oracle, SQL, COM

Language Proficiency:

- Urdu – Mother tongue, English – Proficient, German – Advanced (B1 level/2016)

Software Engineering has become one of the major foci of Computer Science research in Kaiserslautern, Germany. Both the University of Kaiserslautern's Computer Science Department and the Fraunhofer Institute for Experimental Software Engineering (IESE) conduct research that subscribes to the development of complex software applications based on engineering principles. This requires system and process models for managing complexity, methods and techniques for ensuring product and process quality, and scalable formal methods for modeling and simulating system behavior. To understand the potential and limitations of these technologies, experiments need to be conducted for quantitative and qualitative evaluation and improvement. This line of software engineering research, which is based on the experimental scientific paradigm, is referred to as 'Experimental Software Engineering'.

In this series, we publish PhD theses from the Fraunhofer Institute for Experimental Software Engineering (IESE) and from the Software Engineering Research Groups of the Computer Science Department at the University of Kaiserslautern. PhD theses that originate elsewhere can be included, if accepted by the Editorial Board.

Editor-in-Chief: Prof. Dr. Dieter Rombach

Executive Consultant & Founding Director of Fraunhofer IESE and Head of the AGSE Group of the Computer Science Department, University of Kaiserslautern

Editorial Board Member: Prof. Dr. Peter Liggesmeyer

Director of Fraunhofer IESE and Head of the SEDA Group of the Computer Science Department, University of Kaiserslautern

Editorial Board Member: Prof. Dr. Frank Bomarius

Deputy Director of Fraunhofer IESE and Professor for Computer Science at the Department of Engineering, University of Applied Sciences, Kaiserslautern

ISBN 978-3-8396-1568-3

