

# **Verbesserte Software Kostenschätzung - Eine robuste und interpretierbare Modellierungsmethode und deren umfassende empirische Überprüfung**

Isabella Wieczorek

Fraunhofer Institut Experimentelles Software Engineering  
Sauerwiesen 6  
D-67661 Kaiserslautern  
Isabella.Wieczorek@iese.fhg.de

Software Produkte im Zeit- und Kostenrahmen und in akzeptabler Qualität auszuliefern, ist essentiell für viele Organisationen. Aufgrund vieler Probleme in der Praxis wird allerdings nur bedingt verstanden wie valide Kostenvorhersagen zu ermitteln und zu interpretieren sind. Diese Probleme werden durch gängige Kostenschätzmethoden nur unzureichend adressiert, was zu deren geringer Akzeptanz in der Praxis führt.

Diese Dissertation schlägt die Methode Optimized Set Reduction (OSR<sup>+</sup>) vor, um Schätzprobleme im Rahmen von Softwareentwicklungsprojekten zu lösen. OSR<sup>+</sup> basiert auf statistischer Modellierung, generiert aber einfach zu interpretierende, regelbasierte Modelle. Die Vorteile von OSR<sup>+</sup> wurden umfassend in zwei Fallstudien und durch eine subjektive Einschätzung evaluiert. OSR<sup>+</sup> liefert Vorhersagen vergleichbar mit klassischen Verfahren, ist robust gegen Ausreißer, und erzielt bei fehlenden Datenpunkten genauere Ergebnisse als vergleichbare Verfahren.

## **1 Einleitung**

Um wettbewerbsfähig zu bleiben, müssen viele Organisationen heutzutage ihre Softwareprodukte sowohl im Zeit- und Kostenrahmen, als auch in akzeptabler Qualität ausliefern. Genaue Schätzungen sind essentiell zur besseren Planung, Verfolgung und Kontrolle von Softwareprojekten und ebnen den Weg zur erfolgreichen Produktauslieferung.

Software Kostenschätzung muß systematisch unterstützt werden, da viele Organisationen mit engen Terminplänen arbeiten und oftmals ihre Produkte nicht termingerecht und unter hohen Aufwandsüberschreitungen fertig stellen; falls es überhaupt zur Fertigstellung kommt.

Die Ursachen für Termin- und Aufwandsüberschreitungen liegen in vielen praktischen Problemen, im Bereich der notwendigen Messungen und der Modellierung von Zusammenhängen über Softwarekosten. Diese Schwierigkeiten werden nur unzureichend durch gängige Schätzverfahren berücksichtigt, was dazu führt, dass systematische und wiederholbare Verfahren nur bedingt in der Praxis akzeptiert und angewendet werden.

Die Entwicklung geeigneter Kostenschätzverfahren für die Softwareentwicklung hat eine Vielzahl von Methoden hervorgebracht, deren Konzepte aus den verschiedensten Bereichen stammen, wie z.B. Statistik [SSS86, Kit98], Maschinelles Lernen [BFOS84, SS97], oder Wissensakquisition [Hug96]. Trotz dieser Vielfalt, ist die adäquate Nutzung der Verfahren in der Praxis nicht einfach, was oft zu ungenauen Ergebnissen führt.

## 2 Grenzen gängiger Kostenschätzverfahren

Viele statistische, datenbasierte Verfahren liefern immer noch schwer zu interpretierbare Modelle. Die erzeugten Kostenschätzwerte sind jedoch nutzlos ohne ein genaues Verständnis über deren Vorhersagegenauigkeit und deren Gültigkeit und Anwendbarkeit bei auftretenden Projektrisiken. Schlimmer noch, die schwer zu interpretierbaren Resultate zwingen die meisten Praktiker auf black-box Resultate zu vertrauen. Stensrud und Myrteit berichten z.B., dass die Teilnehmer an ihrem Experiment, trotz einer klaren Verbesserung durch regressions-basierte Modelle, nicht von der Nützlichkeit solcher Verfahren überzeugt waren [MS99]. Dies zeigt, dass erst einige psychologische Hindernisse überwunden werden müssen, wenn statistische Verfahren erfolgreich in die Praxis eingeführt werden sollen.

Die Interpretations-Schwierigkeiten hängen sehr stark mit der Komplexität der Modelle zusammen. Je mehr Variablen in einem Modell verwendet werden, desto schwieriger die Interpretation. Um dies zu reduzieren, würde man normalerweise zwischen den Hauptkostentreibern und den Variablen, die nur einen indirekten Einfluss im Kontext von anderen Variablen haben, unterscheiden wollen.

Neben der Interpretierbarkeit von Modellen, sollten empirisch-basierte Schätzverfahren verschiedene Randbedingungen, die mit Software-Engineering Daten zusammenhängen berücksichtigen [BBT92, GM97]. Die Forschungsarbeiten in diesem Bereich berichten von einigen typischen Einschränkungen bezüglich der Analyse und der praktischen Verwendung von Software-Engineering Daten.

- *Annahmen über Projektattribute:* Normalerweise ist es sehr schwer valide Vorannahmen über funktionale Zusammenhänge von Projektattributen und deren Verteilungen zu machen. Dies liegt daran, dass Daten oft nur in geringen Mengen vorhanden sind und große Varianzen in den Daten auftreten. Unter diesen Randbedingungen sind parametrische Verfahren nur begrenzt einsetzbar. Ein Verfahren sollte im Idealfall viele Vorannahmen über Zusammenhänge in den Daten vermeiden.
- *Behandeln verschiedener Projektattribute:* Software-Engineering Datenbanken umfassen im Allgemeinen eine Kombination von Projektattributen (Variablen) verschiedenen Typs (diskret und kontinuierlich). Die Anwendung klassischer Verfahren ist auf kontinuierliche (z.B. interval skalierte) oder diskrete (z.B. nominal skalierte) Variablen begrenzt. Um nun Daten effektiv analysieren zu können, sind Modellierungsverfahren gefragt, die gleichermaßen kontinuierliche wie diskrete Variablen in ihren Modellen verwerten.

- *Robustheit gegenüber Ausreißern:* Software-Engineering Daten enthalten üblicherweise Datenpunkte, die nicht den generellen Trend widerspiegeln. Dies kann z.B. dadurch begründet sein, dass Daten nicht genau erfasst werden können, oder durch das Auftreten von unvorhergesehenen Ereignissen während der Projektentwicklung. Es ist erstrebenswert zwischen typischen und Ausreißerdatenpunkten zu unterscheiden um valide Resultate zu erlangen. Im Allgemeinen sind Verfahren, die robust gegen Ausreißer sind vorzuziehen.
- *Unterschiedlich starke Effekte von Variablen:* Die Relevanz eines Kostentreibers kann abhängig vom Wertebereich sehr variieren. Dies kann unerwünschte Effekte auf die Genauigkeit eines Modell haben. Klassische Regressionsverfahren gehen jedoch von einer durchschnittlich konstanten Relevanz eines Kostentreibers aus.
- *Probleme mit fehlenden Datenpunkten:* Fehlende Daten sind ein generelles Problem bei der Erhebung von Software Engineering Daten [BBT92, GM97]. Dafür gibt es eine Vielzahl von Gründen, wie z.B. keine rechtzeitige Erhebung von Daten, oder Vernachlässigung der Datensammlung aufgrund von Zeitdruck während des Projekts, oder Auftreten technischer Probleme bei der Datensammlung. Der einfachste Weg mit fehlenden Datenpunkten umzugehen ist es, die Projekte (Datensätze) mit fehlenden Daten zu ignorieren. Dies kann allerdings zur Verwerfung vieler wertvoller und teuer gesammelter Daten führen. Um adäquat mit fehlenden Daten umzugehen, sollten Verfahren die wünschenswerte Eigenschaft besitzen so viele der existierenden Daten wie möglich zu berücksichtigen.
- *Unsicherheit in den Vorhersagen:* Aussagen über die Zuverlässigkeit einer Vorhersage basierend auf vergangenen Daten machen zu können ist ein wichtiger Aspekt in der Softwareentwicklung. Dies kann auch helfen die fehlende Information, die durch fehlende Datenpunkte zustande kommt besser einzuschätzen. Deshalb werden Modelle benötigt, die Aufschluss über die Unsicherheit für jede individuelle Vorhersage geben können.

Der richtige Umgang mit solchen Randbedingungen hat einen großen Einfluss auf die Vorhersagegenauigkeit eines Verfahrens. Bisher existierende und weit verbreitete Verfahren sehen nur begrenzt Mechanismen vor, die einzeln die aufgezeigten Probleme adressieren. Dies verdeutlicht die Notwendigkeit Methoden zu entwickeln, die einfach zu interpretierbare Modelle liefern und trotzdem robust gegenüber den Einschränkungen bei Software-Engineering Daten sind.

Um die limitierten Anwendbarkeit existierender Schätzverfahren zu adressieren, wird im Rahmen dieser Dissertation eine Erweiterung des Verfahrens Optimized Set Reduction (OSR<sup>+</sup>) vorgeschlagen, um Software Kostenschätzprobleme zu lösen [Wie02].

### 3 Das OSR<sup>+</sup> Verfahren

Im Bereich Maschinelles Lernen, "t lernen"t Algorithmen aus Erfahrungen und wenden das Gelernte auf neue Situationen an. Dabei wird die Leistungsfähigkeit der Algorithmen

men immer mehr verbessert. Die Art und adäquate Auswahl der Erfahrung hat hierbei einen großen Einfluss auf den Erfolg dieser Algorithmen. Software Engineering Daten repräsentieren hierbei eine Art von Erfahrungen mit bestimmten Randbedingungen, was zu Problemen beim Einsatz und Modellbildung klassischer, statistischer Verfahren zur Aufwandsvorhersage führt (siehe Kapitel 2).

Ziel dieser Dissertation ist es ein robustes Schätzverfahren bereitzustellen, das die Hauptnachteile anderer Methoden überwindet und Modelle liefert mit einer Vorhersagegenauigkeit, die vergleichbar mit den gängigen Verfahren ist. Das ursprüngliche OSR Verfahren wurde zur Lösung von Klassifikationsproblemen entwickelt und basiert auf maschinellem Lernen und robusten statistischen Verfahren [BBT92, Jor95]. Das Verfahren nutzt die Rigorosität statistischer Modellierung aus und generiert dennoch einfach zu interpretierbare, regel-basierte Modelle. Diese Forschungsarbeit erweitert den OSR Algorithmus (OSR<sup>+</sup> genannt) um Kostenschätzprobleme zu lösen.

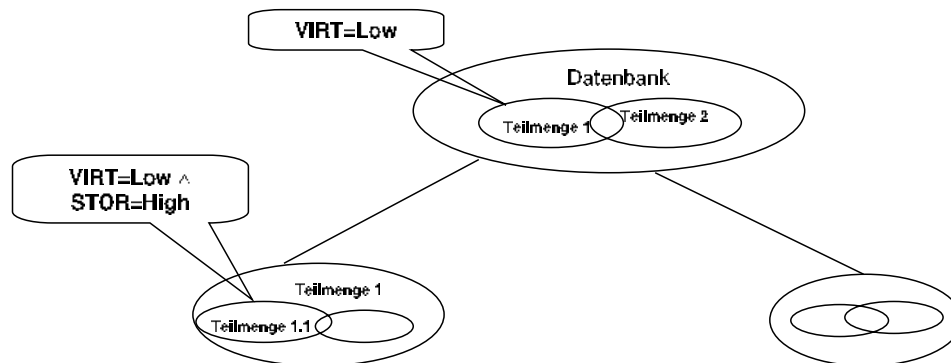
Der Algorithmus selektiert rekursiv Teilmengen aus einer Datenbank, die ein einzuschätzendes Projekt am besten, "t optimal"t charakterisieren. Die Teilmengen werden durch Modelle in Form logischer Ausdrücke beschrieben, die Trends in den Daten repräsentieren. Diese Trends sind nur relevant für die aktuell vorliegende Schätzaufgabe. OSR<sup>+</sup> generiert dynamisch Vorhersagen, die auf das jeweilige Projekt optimal angepasst sind, wodurch die Daten effizient genutzt werden. Die Teilmengenbildung (Datenreduktion) wird also durch die speziellen Charakteristiken des einzelnen, vorherzusagenden Projekts bestimmt. Dies ist ein Unterschied zu anderen Modellierungsverfahren, wie Regression oder Regressionsbäume, wo ein einziges Modell basierend auf den zugrundeliegenden Daten (Trainings-Daten) generiert wird.

Die Teilmengenbildung erzeugt hierbei keine Partitionen, sondern eine Hierarchie, so dass Teilmengen überlappen können. Optimale Teilmengen weisen optimale Wahrscheinlichkeitsverteilungen für den Wertebereich der Vorhersagevariablen auf. Innerhalb solcher Teilmengen konzentriert sich also eine große Anzahl von Projekten auf einen kleinen Wertebereich der Vorhersagevariablen.

Ein Beispiel der Datenreduktion ist in Abbildung 1 dargestellt. Dieses Modell besteht aus einer Konjunktion von zwei COCOMO Kostenfaktoren (virtual machine volatility, main storage constraint) [BAB<sup>+</sup>00]. Der logische Ausdruck charakterisiert in diesem Beispiel den optimalen Datensatz "t Teilmenge 1.1"t.

Optimale Teilmengen werden in einem rekursiven, schrittweisen Prozess generiert, wobei in jedem Schritt die Vorhersagekraft signifikant verbessert wird. Die Hauptgründe für diese Implementierung sind:

- Die Anzahl der möglichen Variablenkombinationen kann zu untragbaren Laufzeiten führen. Deshalb ist eine Suchheuristik notwendig.
- Neben den Teilmengen sind auch die Prädikate, die diese Mengen charakterisieren von Interesse. Die Prädikate sollten hierbei nur Variablen enthalten, die einen signifikanten Einfluß auf die Vorhersage haben enthalten, wodurch sich die Interpretierbarkeit erhöht.

Abbildung 1: Die OSR<sup>+</sup> Datenreduktion

- Es sollte möglich sein Bedingungen, unter denen bestimmte Variablen einen Einfluß haben zu unterscheiden.
- Der relative Einfluß verschiedener Variablen während des Datenreduktionsprozesses sollte bewahrt bleiben.

#### 4 Evaluierung von OSR<sup>+</sup>

Um die Vorteile von OSR<sup>+</sup> zu evaluieren beschreibt diese Dissertation eine subjektive Einschätzung und einen umfassenden empirischen Vergleich der Vorhersagegenauigkeit von OSR<sup>+</sup> mit verschiedenen gängigen Verfahren.

Da der Vergleich der Vorhersagegenauigkeit von OSR<sup>+</sup> externe Validität gewährleisten soll, sind die Umstände unter denen die Evaluierung durchgeführt wird von essentieller Bedeutung. In der Vergangenheit wurden viele vergleichende Studien über die Genauigkeit von Software Kostenschätzverfahren durchgeführt [BW02]. Trotz der intensiven Forschungen könnten nur einige wenige übergreifende Schlussfolgerungen aus den existierenden Resultaten gezogen werden [MK92, WJ99, HCYM98].

Ein Grund dafür liegt darin, dass nur kleine bzw. mittlere Datenbanken zur Durchführung der Studien zur Verfügung standen. Zusätzlich sind bisherige Studien nur unvollständig, da sie immer nur einen kleinen Teil existierender Verfahren miteinander vergleichen [BM95, SS97]. Außerdem wurden bisher nur wenige Studien repliziert. Verfahren wurden nur selten mit Daten validiert, die nicht im Kontext der original eingeführten Verfahren verwendet wurden [CM96]. Auch wenn über mehrere Studien hinweg die gleichen Daten zu Grunde gelegt wurden, so waren die Resultate nicht unbedingt vergleichbar, da verschiedene experimentelle Entwürfe verwendet wurden. Wie in jedem anderen experimentellen Umfeld ist die Replikation ein Schlüsselfaktor um eine Validität der Ergebnisse zu erreichen und konsistente Trends in den Resultaten aufzudecken.

Um den Schwächen vieler vergangener Studien entgegenzuwirken, werden in dieser Arbeit Kostenschätzverfahren konsistent und umfangreich evaluiert. Dies beinhaltet den Vergleich von OSR<sup>+</sup> mit gängigen Verfahren unter Verwendung großer Software-Engineering Datenbanken aus verschiedenen Anwendungsbereichen.

Die zum Vergleich von OSR<sup>+</sup> herangezogenen empirisch-basierten Schätzverfahren, wurden aufgrund bestimmter Selektionskriterien ausgewählt, wie z.B., die Anwendung der Verfahren im Software Engineering Kontext, die Automatisierbarkeit der Verfahren, oder die Anforderungen an die Eingabeparameter. Neben OSR<sup>+</sup> erfüllten folgende Verfahren die Kriterien: lineare Regression [SSS86], sowie robuste Regressionsverfahren [RL87], Varianzanalyse [Kit98], Regressionsbäume [BFOS84], und Analogie-basierte Ansätze zur Kostenabschätzung [SS97]. Kombinationen der obigen Verfahren wurden ebenfalls eingesetzt, um einen potentiell verbesserte Genauigkeit zu testen [Jor95].

Im Rahmen dieser Arbeit wurden zwei große Datenbanken verwendet. Diese Datenbanken enthalten aufwandsbezogene Daten verschiedener Organisationen aus Europa und wurden mittels standardisierter Definitionen der Daten und Nachbearbeitungsverfahren gesammelt. Eine Datenbank enthält Daten von Softwareprojekten der European Space Agency (ESA), die hauptsächlich aus der Luft und Raumfahrt Anwendungsdomäne stammen. Die zweite Datenbank stammt aus Finnland und enthält Daten über Management-Informationen-Systemen (gennant Laturi). Beide Datenbanken sind das Resultat von rigoroser Qualitätssicherung der Daten und werden professionell verwaltet. Die ESA Daten werden von einer internationalen Management Schule in Frankreich namens INSEAD verwaltet [Ins]. Die Laturi Daten werden von der Software Technology Transfer Finland (STTF) verwaltet [STT]. Beide Datenbanken sind, mit zwischen 160 und 206 Projekten, relativ groß für Software-Engineering Daten.

Neben der Vorhersagegenauigkeit existieren weitere wichtige Eigenschaften von Schätzverfahren, die jedoch sehr selten in Studien evaluiert werden. Wenn jedoch Schätzverfahren für die Praxis umfassend evaluiert werden, sind viele Kriterien relevant um geeignete Entscheidungen über den Einsatz von Schätzverfahren treffen zu können. Diese Kriterien beinhalten z.B., die Komplexität eines Verfahrens oder die Interpretierbarkeit der erstellten Schätzmodelle. Einige dieser Kriterien sind nur subjektiv einschätzbar aber dennoch wichtig. Deshalb werden in dieser Arbeit subjektiv die betrachteten Verfahren entsprechend der Kriterien evaluiert. Der Ansatz hierbei ist die Einschätzung der Verfahren unter Berücksichtigung der zugrundeliegenden Annahmen, Stärken und Schwächen. Dies beinhaltet die Definition eines Rahmenwerks zur subjektiven Evaluierung.

Die verwendeten Datenbanken zur Evaluierung erlauben es, Schätzverfahren basierend auf Daten, die von einer Organisation stammen und basierend auf Daten über Organisationen hinweg zu beurteilen. Lokale, aus einer Organisation stammende Daten werden als eine gute Basis für genaue Schätzungen angesehen. Die lokale, firmenspezifische Datensammlung hat verschiedene Vorteile, wie z.B. eine Kontrolle über die Datensammlung durch den Zugriff auf die Datenlieferanten, oder die Möglichkeit zur Definition von Kostenfaktoren, die zugeschnitten sind auf die Organisation [JRW00]. Andererseits existieren in der Praxis oftmals keine Daten über vergangene Projekte, weil die Sammlung von Daten teuer und zeitaufwändig für eine Organisation ist.

Datenbanken, die verschiedene Organisationen beinhalten, bieten hingegen die Möglichkeit der schnellen Datenansammlung. Andererseits kann unter Umständen die konsistente Datensammlung schwer werden, und die Trends in den Daten können sich signifikant über Organisationen hinweg unterscheiden.

Falls Schätzmodelle, die auf firmenübergreifenden Daten basieren vergleichbare Ergebnisse liefern mit Modellen die auf firmeninternen Daten basieren, so hat dies Implikationen für die strategische Ausrichtung von Unternehmen.

Deshalb wurde im Rahmen dieser Arbeit auch der potentielle Vorteil lokaler Datensammlung gegenüber den Vorteilen von Daten von mehreren Organisationen für die Kostenschätzung gegeneinander abgewogen [BEM<sup>+</sup>99, BLW00]. Die dahinterliegende analytische Fragestellung lautet: Ist die Vorhersagegenauigkeit von Schätzverfahren höher, wenn lokale, firmeninterne Daten herangezogen werden, oder wenn firmenübergreifende Daten verwendet werden?

Die Resultate der empirischen Vergleichs von OSR<sup>+</sup> mit gängigen Verfahren zeigen, dass OSR<sup>+</sup> genauere Vorhersagen liefert, als vergleichbare nicht-parametrische Verfahren (Regressions bäume, Analogie-basierte Verfahren). OSR<sup>+</sup> generiert Resultate, vergleichbar mit parametrischen Verfahren (Regressionsverfahren, Varianzanalyse). OSR<sup>+</sup> ist robuster gegenüber Ausreißern als alle anderen berücksichtigten Verfahren. Bei fehlenden Datenpunkten kann OSR<sup>+</sup> durch die effiziente Ausnutzung der vorhandenen Daten mehr und genauere Vorhersagen treffen als vergleichbare Schätzmethoden. In Situationen also, wo die Interaktion zwischen Experten wichtig für die Interpretation von Kostenmodellen ist, kann OSR<sup>+</sup> als gute und sogar bessere Alternative zu parametrischen und nicht-parametrischen Schätzverfahren herangezogen werden.

Die Ergebnisse zur Abwägung lokaler gegenüber firmenübergreifender Datensammlung als Basis zur Vorhersage von Softwarekosten waren überraschend. Werden standardisierte Kostenfaktoren gesammelt, so weisen die Vorhersageergebnisse keinen signifikanten Unterschied zwischen firmeninterner und firmenübergreifender Datensammlung auf. Um wirklich von firmenspezifischer Datensammlung zu profitieren, sollte eine Organisation deshalb auf die wichtigen Kostenfaktoren fokussieren. Diese firmenspezifischen Faktoren können durch ein zugeschnittenes Messprogramm ermittelt werden [BDR96, Wie97]. Wenn innerhalb einer Firma nur wenige Daten zur Verfügung stehen, könnten gemeinsame, firmenübergreifende Datenrepositories einheitlicher Anwendungsdomänen als eine nützliche Alternative angesehen werden.

## 5 Schlußfolgerungen

Im Rahmen dieser Arbeit werden hauptsächlich zwei Probleme untersucht, die aus vergangenen Forschungsarbeiten und Praxiserfahrungen im Bereich Softwarekostenschätzung abgeleitet sind. Erstens, die begrenzte Anwendbarkeit weitverbreiteter, daten-basierter Verfahren zur Softwarekostenschätzung. Zweitens, die Abwägung der Vorteile und Nachteile von firmenübergreifender gegenüber firmeninterner Datensammlung für die Softwarekostenschätzung.

Als potentielle Lösung des ersten Problems wurde die Methode OSR<sup>+</sup> vorgeschlagen. OSR<sup>+</sup> wurde umfassend evaluiert und lieferte vielversprechende Ergebnisse. Das Ergebnis dieser Arbeit ist insbesondere auch eine Weiterentwicklung und Generalisierung des ursprünglichen OSR-Algorithmus. Dies macht die Analyse von Klassifikations- und Regressionsproblemen aller Art mittels OSR<sup>+</sup> möglich.

Zur Lösung der zweiten Fragestellung wurde ein wiederholbarer Vergleich von OSR<sup>+</sup> und gängigen Verfahren durchgeführt, der lokale und firmenübergreifende Daten berücksichtigt.

Die angemessene Auswahl eines Verfahrens ist kontextabhängig und andere Kriterien neben der Vorhersagegenauigkeit sollten ebenfalls berücksichtigt werden. Kriterien wie Komplexität, Interpretierbarkeit, und praktische Überlegungen sind sicherlich maßgeblich. OSR<sup>+</sup> ist eine gute Alternative, da dieses Verfahren einfach zu interpretierbare Modelle liefert, die ein Verständnis der Abhängigkeiten zwischen Kostenfaktoren erlauben und Unsicherheiten in Vorhersagen mitberücksichtigen.

## Literaturverzeichnis

- [BAB<sup>+</sup>00] B.W. Boehm, C. Abts, A.W. Brown, S. Chulani, B.K. Clark, W. Horowitz, R. Madachy, D. Reifer, and B. Steece. *Software Cost Estimation with COCOMO II*. Prentice-Hall, 2000.
- [BBT92] L. C. Briand, V. R. Basili, and W. M. Thomas. A Pattern Recognition Approach for Software Engineering Data Analysis. *IEEE Transactions on Software Engineering*, 18(11):931–42, 1992.
- [BDR96] L. C. Briand, C. Differding, and D. Rombach. Practical Guidelines for Measurement-based Improvement. *Software Process - Improvement and Practice*, 2(4):253–280, 1996.
- [BEM<sup>+</sup>99] L.C. Briand, K. El Emam, K. Maxwell, D. Surmann, and I. Wieczorek. An Assessment and Comparison of Common Cost Software Project Estimation Methods. In *Proceedings of the 21st International Conference on Software Engineering*, pages 313–322, 1999.
- [BFOS84] L. Breiman, J. Friedman, R. Ohlsen, and C. Stone. *Classification and Regression Trees*. Wadsworth & Brooks/Cole Advanced Books & Software, 1984.
- [BLW00] L.C. Briand, T. Langley, and I. Wieczorek. A replicated Assessment and Comparison of Common Software Cost Modeling Techniques. In *Proceedings of the 22nd International Conference on Software Engineering*, pages 377–386, 2000.
- [BM95] R. Bisio and F. Malabocchia. Cost Estimation of Software Projects through Case Based Reasoning. In *International Conference on Case Based Reasoning Research and Development, ICCBR*, pages 11–22, 1995.
- [BW02] L.C. Briand and I. Wieczorek. *Encyclopedia of Software Engineering*, volume 2, chapter Software Resource Estimation, pages 1160–1196. John Wiley & Sons, 2 edition, 2002.



- [CM96] D. Chatzoglou and L.A. Macaulay. A review of existing models for project planning and estimation and the need for a new approach. *International Journal of Project Management*, 14(3):173–183, 1996.
- [GM97] A.R. Gray and S.G. MacDonell. A comparison of techniques for developing predictive models of software metrics. *Information and Software Technology*, 39:425–437, 1997.
- [HCYM98] R.T. Hughes, A. Cunliffe, and F. Young-Martos. Evaluating software development effort modelbuilding techniques for application in a real-time telecommunication environment. *IEE Proceedings Software*, 145(1):29–33, 1998.
- [Hug96] R.T. Hughes. Expert judgement as an estimating method. *Information and Software Technology*, 38(2):67–75, 1996.
- [Ins] <http://www.insead.fr>.
- [Jor95] M. Jorgensen. Experience With the Accuracy of Software Maintenance Task Effort Prediction Models. *IEEE Transactions on software Engineering*, 21(8):674–681, 1995.
- [JRW00] D.R. Jeffery, M. Ruhe, and I. Wiecek. A Comparative Study of Two Software Development Cost Modeling Techniques using Multi-organizational and Company-specific Data. *Information and Software Technology*, 42:1009–1016, 2000.
- [Kit98] B. Kitchenham. A Procedure for Analyzing Unbalanced Datasets. *IEEE Transactions on Software Engineering*, 24(4):278–301, 1998.
- [MK92] T. Mukhopadhyay and S. Kerke. Software Effort Models for Early Estimation of Process Control Applications. *IEEE Transactions on Software Engineering*, 18(10):915–924, 1992.
- [MS99] I. Myrtveit and E. Stensrud. A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models. *Transactions on Software Engineering*, 25(4):510–525, 1999.
- [RL87] P.J. Rousseeuw and A.M. Leroy. Least Median of Squares Regression. *Wiley Series in Probability and Mathematical Statistics*, 1987.
- [SS97] M. Shepperd and C. Schofield. Estimating Software Project Effort Using Analogies. *IEEE Transactions on Software Engineering*, 23(12):736–43, 1997.
- [SSS86] L.D. Schroeder, D.L. Sojoquist, and P.E. Stephan. *Understanding Regression Analysis - An introductory Guide*. Series: Quantitative Applications in the Social Sciences, 57, SAGE University Paper, 1986.
- [STT] <http://www.sttf.fi/index.html>.
- [Wie97] I. Wiecek. On the Establishment of successful Measurement Programs in Industry. *Software Process - Improvement and Practice*, 3:191–194, 1997.
- [Wie02] I. Wiecek. *Improved Software Cost Estimation - A Robust and Interpretable Modeling Method and a Comprehensive Empirical Investigation*, volume 7. Fraunhofer IRB Verlag, 2002.
- [WJ99] F. Walkerdien and R. Jeffery. An Empirical Study of Analogy-based Software Effort Estimation. *Empirical Software Engineering*, 4:135–158, 1999.

**Isabella Wiczorek** hat an der Universität Koblenz Informatik studiert und mit dem Diplom 1994 abgeschlossen. Im Jahre 1995 war sie an der Universität Kaiserslautern als wissenschaftliche Mitarbeiterin angestellt. Dort arbeitete sie in anwendungsnahen Transferprojekten in der Arbeitsgruppe Software Engineering, Software Technologie Transfer Initiative (STTI-KL). Im Rahmen dieser Tätigkeiten transferierte sie verschiedene Methoden zur messbasierten Verbesserung in die industrielle Praxis. Seit 1996 arbeitet sie am Fraunhofer Institut Experimentelles Software Engineering (IESE), wo sie zur Zeit die wissenschaftliche Leitung für die Abteilung Qualitäts- und Prozess Engineering hat. Sie hat in den ganzen Jahren in verschiedenen Industrieprojekten im Bereich Software Kosten- und Qualitätsmodellierung gearbeitet. Ihre Hauptinteressen liegen in den Bereichen Software Messen und Bewerten, empirische Modellbildung, und Entwicklung quantitativer Methoden zur Software Prozess Verbesserung. Im Juni 2001 hat Frau Wiczorek ihre Promotion mit Auszeichnung abgeschlossen, wofür sie den Preis "t'Beste Dissertation 2000/2001"t' von der Universität Kaiserslautern erhielt.