

# ARBEITSKREIS SOFTWAREMETRIKEN

*Reiner Dumke*

Der inzwischen 6. Workshop des GI-Arbeitskreises Softwaremetriken wurde mit internationaler Beteiligung gemeinsam mit dem kanadischen Centre d'Intérêt sur les Métriques (CIM) und dem Arbeitskreis Qualität,tsverbesserung der GI-FG 2.1.9 (Objektorientierte Softwareentwicklung) durchgeführt. Der Tagungsort Universität, Regensburg sorgte für ein gutes Ambiente für die zweit,,gigen Vortr,,ge, Diskussionen und Absprachen in den Universit,,sr,,umen und in der Regensburger Altstadt.

Der Workshop widmete sich vor allem den Schwerpunkten

- Metriken für die objektorientierte Softwareentwicklung,
- Softwaremetriken für neue Anwendungsbereiche und -paradigmen,
- Metrikenprogramme bzw. Measurement Frameworks für die industrielle Anwendung (Methoden und Erfahrungen).

Das Programm- und Organisationskomitee bedankt sich an dieser Stelle noch einmal bei Prof. Lehner für die Gew,,hrleistung sehr guter Arbeitsbedingungen. Die Ergebnisse des Workshop sind in einem englischsprachigen Buch mit dem Titel *Software Metrics* beim Gabler-Verlag Wiesbaden zusammengefaát und voraussichtlich ab Ende Juni dieses Jahres im Buchhandel erh,,ltlich.

Im folgenden wird zu einigen ausgew,,hlten Beitr,,gen eine kurze Inhaltsbeschreibung angegeben:

**Ebert, C.:**

*(Alcatel Telecom, Antwerpen, Belgien)* **Applying Knowledge-Based Techniques to Software Quality Management**

Der Autor beschreibt ausgehend von jahrelangen Erfahrungen bei der Anwendung statistischer Methoden für die Metriken- bzw. Meádatenauswertung (wie z. B. der Clusteranalyse, der Diskriminanzanalyse, den Entscheidungs,,umen) die Notwendigkeit, das Expertenwissen des Software Engineering in den Entscheidungs- bzw. Verbesserungsprozeá der Softwareentwicklung einflieáen zu lassen. Dabei wird die Verwendung von fuzzy-basierten Expertensystemen, die teilweise für den PC vorhanden sind, diskutiert und deren besondere

Eignung hervorgehoben. Erste Anwendungen werden an zwei Projektbeispielen (einer Betriebssystem- und einer Netzverteilungssystementwicklung) erläutert.

### **Foshag, O.**

#### *(SES, München)* **Metrics for Evaluation of Maintenance Service**

In diesem Beitrag werden ausgehend von der allgemeinen Problematik der Softwarewartung behauptet sieben Metriken zur Beurteilung der Produktivität und der Qualität der Wartungsdienstleistung aufgestellt:

1. *die Zeit* zur Erfüllung der Wartungsaufträge in Kalendertagen relativ zum Umfang der Wartungsaufträge in betroffenen Code-Anweisungen bzw. Function-Points, Data-Points oder Object-Points;
2. *der Aufwand* zur Erfüllung der Wartungsaufträge in Personentagen relativ zum Umfang der Wartungsaufträge;
3. *die Anzahl der Fehler*, die durch die Erfüllung der Wartungsaufträge entstehen relativ zum Umfang der Wartungsaufträge;
4. *die Qualität und Komplexität* der Programme nach der Erfüllung der Wartungsaufträge relativ zur Qualität und Komplexität der Programme vorher;
5. *die Qualität der Dokumentation* bezüglich der Verständlichkeit, der Genauigkeit und der Übereinstimmung mit den Programmen;
6. *das Verhältnis* der Anzahl der Wartungsprogrammierer zur Softwaregröße in LOC, Anweisungen, Function-Points, Data-Points, Object-Points und/oder Komponenten relativ zur jährlichen Änderungsrate in Prozent geänderter Code-Zeilen;
7. *der Grad der Benutzerzufriedenheit* mit der Softwarewartung bezüglich der Reaktionszeit, den Kosten, dem Nutzen, der Termintreue, der Fehlerrate, der Hilfsbereitschaft und der Verfügbarkeit auf einer Skala von 1 bis 10.

Die einzelne Umsetzung dieser Metriken und deren Tauglichkeit werden an praktischen Beispielen diskutiert.

### **Schmelz, D.:**

#### *(Universität Jena)* **The Use of Factorial Analysis in the Area of Software Metrics**

Bei der Vielzahl heute existierender Softwaremetriken ist es kaum zu übersehen, welche Metriken mehr oder weniger redundant zueinander

sind bzw. wie sich eine mögliche Variabilitätskopplung auf die Interpretation der Messergebnisse auswirkt, die zumeist mehrere Metriken einschließt. Der Autor betrachtet bisher existierende Modelle der Ergebnisinterpretation von Munson, Khoshgoftaar, Robillard u. a. und stellt insbesondere einen Algorithmus zur Modifikation einer Ladungsmatrix für anwenderbezogene Restriktionen zu einzelnen Faktorladungen vor. Durch Vorgabe einer Restriktionsmatrix für eine Ladungsmatrix gegebener Dimension werden unter Ausnutzung des Mehrdeutigkeitsspielraumes der faktoranalytischen Lösung Hypothesen des Anwenders über die Metriken-Faktor-Struktur prüfbar.

**Desharnais, J., Morris, P.:**

*(Total Metrics, Montreal, Kanada)* **Post measurement validation procedure for the Function Point counts**

Die beiden Autoren beschreiben zunächst die beiden grundlegenden Formen des Nachweises der Tauglichkeit und der Qualität einer Softwaremessung als *a priori Validation* und *a posteriori Validation*. Die *a priori Validation* schließt vor allem die Korrektheit der Metriken- bzw. Maßdefinitionen und die richtige methodische Vorgehensweise bei der Software-Messung ein. Die *a posteriori Validation* hingegen dient der Prüfung der tatsächlichen Relevanz der Messdaten, deren Konsistenz und der statistischen Analyse.

Die *a posteriori Validation* wird anhand der Function-Point-Methode für ausgewählte praktische Anwendungen demonstriert. Sie umfaßt die Schritte

1. der dokumentengestützten Vorbereitung der Function-Point-Zählungen,
2. des sogenannten High-Level-Reviews, welches die Bestimmung der für die Function-Point-Methode notwendigen Wichtungsfaktoren und die eigentliche Function-Point-Auszählung einschließt,
3. des Intermediate-Level-Reviews, welches u. a. der Bestimmung der Relationen zwischen der Anzahl der logischen Dateien und den jeweiligen ungewichteten Function-Points, zwischen den einzelnen Funktionen untereinander sowie deren prozentuale Verteilung dient,

4. des Lowest-Level-Reviews, welches die Zuordnung der jeweiligen Function-Points zu den daten- und transaktionsbezogenen Business-Funktionen analysiert,
5. der Dokumentation des Validationsprozesses nach einem vorgestellten Darstellungs- und Beschreibungsstandard.

Die Autoren weisen auf die speziellen Probleme bei der Validation einer Function-Point-Bewertung anhand von Beispielprojekten hin.

**Dumke, R.:**

*(Universität Magdeburg)* **Really Object-Oriented Software Metrics**

Die Anzahl von Software-Metriken für neue Entwicklungs- bzw. Programmierparadigmen steigt sehr schnell an. So gibt es bereits für objektorientierte Programme eine große Anzahl von Umfangs-, Kopplungs-, Kohäsions-, Vererbungs- und anderer Metriken. Der Autor beschreibt zunächst die Measurement Frameworks klassifiziert nach informalen (Capability-Maturity-Modell, ISO 9000 usw.) und formalen (algebraische, axiomatische usw.). Dabei wird auf deren softwaretechnischen Ausprägung als funktionaler Ansatz verwiesen. Wirkliche objektorientierte Softwaremetriken sollten auch dem objektorientierten Ansatz entsprechen. Dieser Grundidee folgend leitet der Autor ein Measurement Framework mit Softwaremetriken her, die die Eigenschaften von Objekten besitzen und als *Softwarequalitätsagenten* in einen Entwicklungsbereich wirken. Erste Anwendungen werden für Java und Smalltalk diskutiert.

**Mittelmann, A.:**

*(Voest-Alpine Stahl, Linz, Österreich)* **Implementation of a Measurement Plan in an Industrial Environment**

Im Rahmen der Umstellung der Softwareentwicklung auf eine objektorientierte Form stellt die Autorin Ansatzpunkte für die Einführung eines Metrikenprogrammes vor. Dabei wird auf die Notwendigkeit der Erarbeitung eines Prozessmodells verwiesen. Als methodische Grundlage wird der AMI-Ansatz (Application of Metrics in Industry) verwendet, der tool-gestützt angewendet werden kann und

- die Prozessbewertung nach einem verfeinerten Capability-Maturity-Modell (CMM),
- die zielebasierte Analyse, Messung und Bewertung und schließlich Prozessverbesserung

beinhaltet. Die Verbesserungsstrategie wiederum basiert auf der Goal-Question-Metric-Methode (GQM). Dabei handelt es um einen zielgerichteten Qualitätsverbesserungsansatz, der es ermöglicht, sich zunächst auf die (selbstgewählten) Schwerpunktbereiche zu konzentrieren. Das AMI-Tool generiert nach jeder CMM-Bewertung und Zielausrichtung einen sogenannten *Measurement Plan*, der die Grundlage für Messungen und darauf basierende Bewertungen darstellt. Der Beitrag enthält erste Erfahrungen bei der Umsetzung der genannten Strategien.

**Fetcke, T., Abran, A., Nguyen, T.:**

*(Quebec-Universität, Montreal, Kanada)* **Mapping the OO-Jacobson Approach to Function Point Analysis**

Die Autoren widmen sich der Problemstellung der Anwendung der Function-Point-Methode für die objektorientierte Softwareentwicklung (OOSE). Insbesondere in den frühen Phasen, in der auch die Function-Points ermittelt werden, unterscheidet sich der objektorientierte Ansatz in dieser Hinsicht von den „klassischen“ Methoden. Die Autoren wählen den OOSE-Ansatz von Jacobson, dessen grundlegendes Konzept das der Use-Cases darstellt. Anhand der durch diese Methode implizierten Daten-, Funktions- und Verhaltensmodelle wird ein möglicher und zweckmäßiger Ansatz für die Anwendung der Function-Point-Methode vorgestellt und an drei ausgewählten Beispielprojekten demonstriert.

**Morschel, I.:**

*(Daimler Benz AG, Forschungszentrum Ulm)* **Quality and Productivity Improvement in Object-Oriented Software Development - Experiences and Lessons Learned**

Der bereits im Arbeitskreis durch zahlreiche Arbeiten zur Messung und Bewertung objektorientierter Programmiersysteme ausgewiesene Autor beschreibt in diesem Beitrag Ansätze zur Verbesserung der Produktivität bei der Softwareentwicklung unter Verwendung der bereits oben genannten GQM-Methode. Dabei werden vor allem Erfahrungen für die objektorientierte Entwicklung in allen Phasen beschrieben und die kritischen Bereiche der Messung, Analyse und Bewertung im eigenen Entwicklungsumfeld erläutert.

**Zuse, H.:**

*(Technische Universität Berlin)* **A Measure Information System**

Für das Gebiet der Software-Messung stellt der Autor ein Lehr- und Lern-Tool vor, das bereits über 1000 Softwaremetriken bzw. -maße und über 1500 Literatureinträge beinhaltet. Dieses Tool (ZD-MIS) besitzt folgende Komponenten

- Auswahlkomponenten für die Softwaremetriken,
- Komponenten für die Literatursuche,
- Function-Point-Berechnungskomponenten,
- Komponenten für die Anzeige spezieller Berechnungen zur Softwaremessung,
- Komponenten zur Bestimmung der minimalen Anzahl von Metriken für einen vorgegebenen Anwendungsbereich,
- Komponenten für die Anleitung zur Einführung eines Metrikenprogrammes in einer Firma.

Die Literatursuche beinhaltet ebenso Funktionen zur Analyse von Referenzen und Quellenangaben der jeweiligen Koautoren. Die Tutorials beziehen sich auf die Grundbegriffe zu den Softwaremetriken, zu Problemen der Software-Messung, zur angewandten Maßtheorie, deren Axiome und den Skalentypen, sowie zu Metrikenarten und deren Validation. All diese Eigenschaften können an den gespeicherten Metriken/Maßen angezeigt und analysiert werden. Ein Sachwortverzeichnis vereinfacht die Suche nach ausgewählten Informationen. Das Tool wird demnächst im Zusammenhang mit dem neuen Buch vom Autor (*A Measurement Framework*) zu erhalten sein.

**Maya, M., Abran, A., Bourque, P.:**

*(Quebec-Universität, Montreal, Kanada)* **Measuring the Size of Small Functional Enhancements to Software**

Die Autoren beschreiben die Quantifizierung des Anteiles der Erweiterungen bei der Softwarewartung mit Hilfe der (erweiterten) Function-Point-Methode. Dabei werden auch Aussagen zur Produktivität und zum jeweiligen Änderungsumfang diskutiert. Die durch die Function-Point-Methode zugrunde gelegte Umfangsbewertung der Erweiterungen erweist sich dabei bezogen auf die vorgestellten praktische Anwendungen nur als grobe Schätzung nutzbar und erfordern die Einbeziehung weiterer Projektmerkmale.

**Hitz, M., Stiller, S.:**

*(Universität Wien, Österreich)* **A Versatile Framework for the Collection of Object-Oriented Software Metrics**

Der sehr umfangreichen Anzahl von Softwaremetriken für die objektorientierte Softwareentwicklung stehen im allgemeinen nur sehr wenige empirische Studien gegenüber, die eine hinreichende Validation dieser Metriken ermöglichen. Um jedoch diese wenigen Erfahrungen effizient zu verwenden, wurde ein allgemeines Tool konzipiert, das einen sprachspezifischen Parser verwendet, der eine allgemeine sprachunabhängige Klassenstruktur erzeugt. Diese ist dann wiederum Grundlage für den jeweiligen Metriken-Kollektor, der zunächst alle „konzepttypischen Metriken“ berechnet und eine rechnergestützte Auswertung mit den empirischen Daten ermöglicht. Das Tool wurde zunächst für C++ implementiert. Erste Anwendungsergebnisse werden vorgestellt.

**Winkler, A., Dumke, R., Koeppe, R.:**

*(Universität Magdeburg)* **Efficiency and Maintainability of Java Applications**

Die Verbreitung neuer Entwicklungs- oder Programmierparadigmen geht im allgemeinen immer mit einigen durchaus überzeugenden Hypothesen einher. So auch bei der Programmiersprache Java. Sie ist nach Meinung der Sprachentwickler „*einfach, robust, sicher, architekturneutral, portabel, in hohem Maße performant*“ usw. Alle Attribute, die aktuelle Probleme der Softwaretechnik ansprechen. Die Autoren verweisen anhand einer allgemeinen Metrikenhierarchie für die Prozeß-, Produkt- und Ressourcenbewertung auf die offenen Probleme in diesem Zusammenhang hin, die insbesondere in der Bewertung der Eigenschaften der Entwicklungsmethodik von Java bestehen. Erste Ansätze für die Anwendung bisheriger Bewertungsformen auf der Grundlage von Softwaremetriken werden aufgezeigt.

**Baisch, E., Ebert, C.:**

*(Alcatel Stuttgart)* **On a Neural-Fuzzy Technique with GA Optimization for Software Quality Models**

Die Autoren verweisen auf die Probleme bei der Anwendung klassischer Qualitätsmodelle, die insbesondere statische Elemente der Metrikdefinition und deren Meáwertauswertung aufweisen. Sie demonstrieren den Einsatz der Neuro-Fuzzy-Technik unter Anwendung

generischer Algorithmen (GA) als eine intelligente Form der Aufwandsschätzung und Softwarebewertung. Die Anwendung wird an einem Telekommunikationsprojekt diskutiert. Als Tool wird der Neural Network Simulator verwendet. Das Beispiel ergab zunächst 10 Eingabeneuronen, 30 Fuzzy-Set-Neuronen und 10 Regel-Neuronen. Dabei konnte gezeigt werden, daß auf diese Weise ca. 79% der als „schlecht“ er- bzw. bekannten Module identifiziert wurden.

# CALL FOR PAPER

*zum 7. Workshop*  
**SOFTWAREMETRIKEN**  
*des Arbeitskreises Softwaremetriken und des  
Arbeitskreises Qualitätsverbesserung*

*Unser diesjähriger Workshop findet in der Zeit vom 18.9. bis 19.9.1997 an der Universität Mannheim statt. Bitte reichen Sie ganze Beiträge (maximal 10 Seiten) oder zwei- bis dreiseitige Abstracts zu folgenden Themen ein:*

- Praktische Erfahrungen beim Einsatz von Softwaremetriken,*
- Konzeptionen und Anwendungen von Erfahrungs-, Meß- und Projektmanagement-Datenbanken,*
- Einsatz von Softwaremetriken fuer die Auswahl von Entwicklungsmethoden (insbesondere objektorientierter Paradigmen),*
- Neue Ansätze der Metrikenvalidation,*
- Stand und Entwicklung von Metriken-Standards.*

*Der Workshop ist keinesfalls auf die genannten Themenstellungen beschränkt und soll darüberhinaus viel Spielraum für Diskussionen und Initiativen geben. Alle Beiträge sollen dabei als Vortrag oder als Auslage (Pinwand) zur Kenntnis gelangen. Die zum Vortrag ausgewählten Beiträge werden in einer geschlossenen Form publiziert.*



Die Beitr.,ge sind bis zum **18. Juli 1997** an eine der folgenden Adressen bzw. per Email zu senden:

- **Kathrin Baumann**, (Leiterin des Arbeitskreises  
Qualit.,tsverbesserung), SAP-AG, Email: [kathrin.baumann@sap-ag.de](mailto:kathrin.baumann@sap-ag.de)
- **Prof. Franz Stetter**, (Workshop-Organisation), Universit.,t Mannheim,  
A5, 68131 Mannheim, Email: [fstetter@pi1.informatik.uni-mannheim.de](mailto:fstetter@pi1.informatik.uni-mannheim.de)
- **Prof. Reiner Dumke**, (Leiter des Arbeitskreises Softwaremetriken),  
Universit.,t Magdeburg, Fakult.,t fr Informatik, Postfach 4120,  
39016 Magdeburg, Email: [dumke@irb.cs.uni-magdeburg.de](mailto:dumke@irb.cs.uni-magdeburg.de)