Softwaremessung und -metrik

AW1 Votrag - Daniel Wojtucki

Hamburg, 20. Januar 2010

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Inhalt

Motivation der Themenwahl

- Bachelorarbeit im Bereich Test- und Qualitätsmanagement
- Vertiefung des Themas
- wichtiges Werkzeug im Software Engineering
- wenig Anwendung in Unternehmen
- möglicher Bestandteil eines Softwarecockpits (Sarstedt)

Probleme des Qualitätsmanagements

- aus Fertigung $\longrightarrow$ Übereinstimmung von Produkt mit Spezifikationen
- Spezifikationen nicht eindeutig
- andere Eigenschaften unberücksichtigt (Wartungseigenschaften)
- Korrelation zwischen Prozess und Produkt nicht eindeutig
- Standards kein Allheilmittel $\longrightarrow$ Entwicklung ist kreativer Prozess
- jeder ist für Produktqualität verantwortlich

Qualitätssicherung

- Softwarestandards (Trial-And-Error-Verfahren)
- Standards aus ISO 9000
- Standards für Dokumente
- Qualitätskontrolle hauptsächlich durch Reviews
  - Entwurfs- oder Programminspektionen
  - Fortschritts-Reviews
  - Qualitäts-Reviews

Definition Softwaremetrik

- Analysetechnik zur Qualitätssicherung auf bereits erstellten Entitäten
- Messen von Merkmalen eines Softwareprodukts
- Ableitung eines numerischen Werts
- Messung jeder Art bei
  - Softwaresystemen
  - Prozessen
  - dazugehörigen Dokumenten
- Definition nach IEEE Standard 1061

Motivation für den Einsatz von Metriken

- beschleunigtes Verfahren statischer Qualitätssicherung benötigt
  - Qualitäts-Reviews aufwändig und langsam
  - Verzögerung im Softwareprozess
- mess- und vergleichbare Qualität der SW
- Möglichkeit zur Überprüfung einer Prozessverbesserung
- durch Vergleiche können Abschätzungen erfolgen

Kategorisierung

- Prozessmetriken (für Aktivitäten bei Bearbeitung)
- Ressourcenmetriken (für Prozessmittel)
- Produktmetriken (für Artefakte als Prozessergebnis)
  - dynamische Metriken (bspw. Effizienz, Zuverlässigkeit)
  - **statische Metriken**

Weitere Unterteilung in:

- direkte Metrik (keine Abhängigkeit)
- indirekte Metrik (indirekte Messung über andere Metrik oder Entität)

Vorraussetzungen einer korrekten Messung

Gütekriterien von Metriken

- Objektivität (keine subjektiven Einflüsse)
- Robustheit (wiederholbare Messungen)
- Vergleichbarkeit (gleiche Kenngrößen müssen vergleichbar sein)
- Ökonomie (kostenökonomisch)
- Korrelation (solider Rückschluss auf die überwachte Kenngröße, bspw. Laufzeiteffizienz)
- Verwendbarkeit (kein Selbstzweck!)

Prozess des Messvorgangs

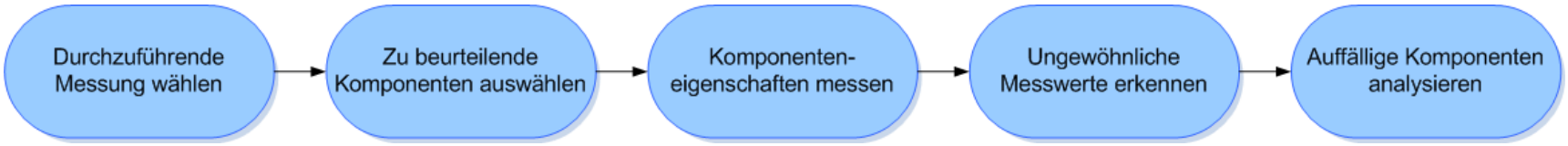


Abbildung: Prozess der Produktmessung nach [Sommerville2007]

- Auswahl der Metriken (bspw. nach GQM-Muster)
- Auswahl der Komponenten (kritische oder repräsentative)
- eigentliche Messung (CASE-Tools oder Eigenentwicklungen)
- ungewöhnliche Messwerte erkennen (hohe oder niedrige Werte, vergleichen)
- auffällige Komponenten analysieren (Qualität beeinträchtigt?)

Goal-Question-Metric Muster

- Warum soll gemessen werden? ⟶ **Goal**
- Was soll gemessen werden? ⟶ **Question**
- Wahl der Messung ⟶ **Metric**

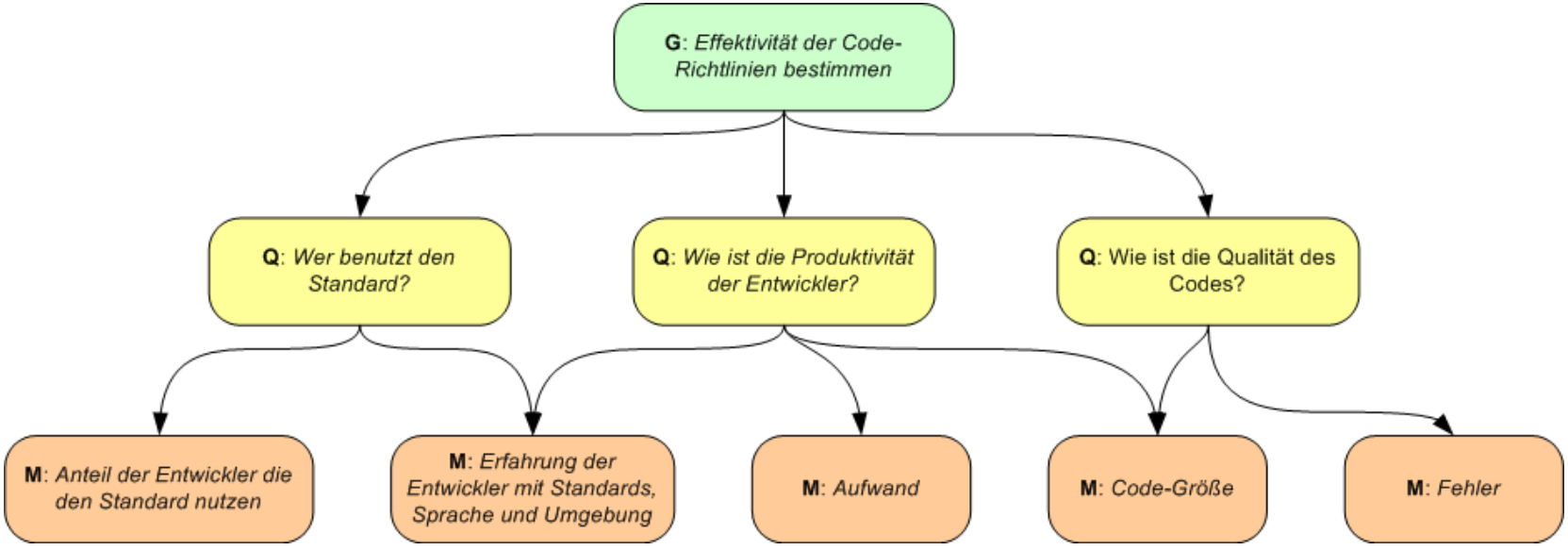


Abbildung: Beispiel eines möglichen GQM-Graph nach [Koschke2006]

LOC, NCSS und Co.

- **LOC** $\longrightarrow$ *Lines of Code*
  - Variation ohne Leerzeilen
- **NCSS** $\longrightarrow$ *Non Commented Source Statements*
- weitere Variationen:
  - alle Zeilen mit ausführbarem Code
  - alle Zeilen mit ausführbarem Code und Variablendeklaration
  - alle Programmsymbole (*token*)
  - alle Anweisungsseparatoren (z.B. Semikolon in Java)
- einfache Volumenmetrik $\longrightarrow$ für Komplexität ungeeignet
- **Basisparameter**, Verbreitung in Firmen
- Sprachfaktoren zur Gewichtung

```
624             }
625 } else {
626 // we're connected
627 InetAddress packetAdd
628 packetAddress = p.get
629 if (packetAddress ==
630     p.setAddress(conn
631     p.setPort(connect
632 } else if ((!packetAd
633         p.getPort() !=
634     throw new Securit
635 " differ");
636 }
637 }
638         byte dttl = g
639         try {
640             if (ttl !
641                 // se
642     getImpl().setTTL(
643 }
644 // call the datagram
645     getImpl().send(p)
646 } finally {
647     // set it back to
648         if (ttl != dt
649             getImpl()
650         }
651     }
652 } // synch p
653 }  //synch ttl
654 } //method
655 }
656
```

Halstead-Metriken - Teil 1

- lexikalische Metrik (Maurice Howard Halstead 1977)
- "*Zählen von Zeichen*" (Operatoren und Operanden)
- Zusammenhang zwischen Programmkomplexität und Auftrittscharakteristik lexikalischer Elemente
- unabhängig von Formatierung (anders als LOC)
- Programmiersprachen abhängig

Halstead-Metriken - Teil 2

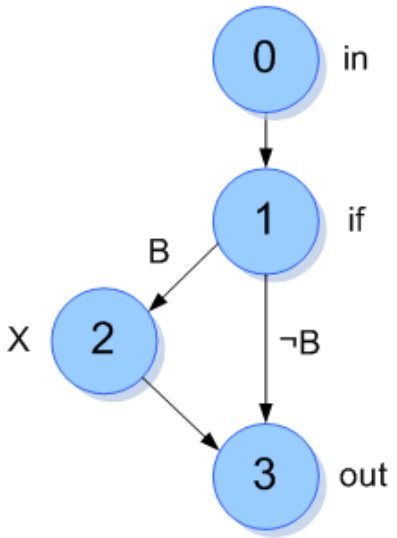
Übersicht einiger Halstead-Metriken:

- $\eta_1 =$ Anzahl unterschiedlicher Operatoren
- $\eta_2 =$ Anzahl unterschiedlicher Operanden
- $N_1 =$ Gesamtzahl der Vorkommen aller Operatoren
- $N_2 =$ Gesamtzahl der Vorkommen aller Operanden
- Vokabular ($\eta = \eta_1 + \eta_2$)
- Minimalvokabular ($\eta^* = \eta_2^* + 2$)
- Länge ($N = N_1 + N_2$)
- Volumen ($V = N \cdot log_2 \eta$)
- Minimalvolumen ($V^* = \eta^* \cdot log_2 \eta^*$)
- Level ($L = \frac{V^*}{V}$)

McCabe-Metrik

- Komplexität von SW-Komponente
- Graphentheorie $\longrightarrow$ *zyklomatische Zahl*
- ***zyklomatische Komplexität***
- $v(G) = |E| - |N| + 2$
- meist auf Funktionsebene
- obere Grenze von 10 nach McCabe
- für Programmkomplexität gilt: $\sum_{i=1}^{n} v(G_i)$
- Programmierverhalten der Entwickler

```
if (B) X;
```



|E| = 4
|N| = 4

v(G) = 4 - 4 + 2 = 2

Warum objektorientierte Metriken?

- klassische Metriken für *imperative Programmierung*
- nur auf Methodenimplementierung aussagekräftig anwendbar
  - Methoden sind imperativ
- Besonderheiten der OO nicht berücksichtigt:
  - Klassenbeziehungen
  - Vererbungsmechanismus

Komponentenmetriken

- separate Bewertung der Bestandteile (Klasse, Paket)
- hohe Signifikanz für Qualitätsbewertung
  - empirische Untersuchungen
- Hilfsmittel für Refactoring

Abkürzung	Metrik	Typ
OV	Object Variables	Umfangsmetrik
CV	Class Variables	Umfangsmetrik
NOA	Number of Attributes	Umfangsmetrik
WAC	Weighted Attributes per Class	Umfangsmetrik
WMC	Weighted Methods per Class	Umfangsmetrik
DOI	Depth Of Inheritance	Vererbungsmetrik
NOD	Number of Descendants	Vererbungsmetrik
NORM	Number of Redefined Methods	Vererbungsmetrik

Tabelle: Auswahl an Komponentenmetriken nach [Hoffmann2008]

Komponentenmetriken

- separate Bewertung der Bestandteile (Klasse, Paket)
- hohe Signifikanz für Qualitätsbewertung
  - empirische Untersuchungen
- Hilfsmittel für Refactoring

Abkürzung	**Metrik**	**Typ**
OV	Object Variables	Umfangsmetrik
CV	Class Variables	Umfangsmetrik
NOA	Number of Attributes	Umfangsmetrik
WAC	Weighted Attributes per Class	Umfangsmetrik
WMC	Weighted Methods per Class	Umfangsmetrik
DOI	Depth Of Inheritance	Vererbungsmetrik
NOD	Number of Descendants	Vererbungsmetrik
NORM	Number of Redefined Methods	Vererbungsmetrik

Tabelle: Auswahl an Komponentenmetriken nach [Hoffmann2008]

Strukturmetriken

- Betrachtung des Klassenverbunds als Ganzes
- Analyse von Interaktions- und Vererbungsmustern

- Basis-Metriken für Kopplungsanalyse
  - **Fan-In**: Anzahl zugreifender Klassen $F_{in}(C)$
  - **Fan-Out**: Anzahl zugegriffener Klassen $F_{out}(C)$
- strukturelles Komplexitätsmaß für Klasse C
  - $v_{HK}(C) = (F_{in}(C) \cdot F_{out}(C))^2$
- Erweiterung der Definition
  - $V_{HS}(C) = v_{internal}(C) \cdot (F_{in}(C) \cdot F_{out}(C))^2$

Strukturmetriken

- Betrachtung des Klassenverbunds als Ganzes
- Analyse von Interaktions- und Vererbungsmustern

- Basis-Metriken für Kopplungsanalyse
  - **Fan-In**: Anzahl zugreifender Klassen $F_{in}(C)$
  - **Fan-Out**: Anzahl zugegriffener Klassen $F_{out}(C)$
- strukturelles Komplexitätsmaß für Klasse C
  - $v_{HK}(C) = (F_{in}(C) \cdot F_{out}(C))^2$
- Erweiterung der Definition
  - $V_{HS}(C) = v_{internal}(C) \cdot (F_{in}(C) \cdot F_{out}(C))^2$

Zusammenfassung

- wichtiges Werkzeug für Qualitätssicherung
- Unterstützung bei Prozesssteuerung
- das Ziel der Metrik muss klar definiert sein
- wenig Verwendung in der Praxis (Problem: Auswahl der richtigen Metriken)
- Werkzeug für die Erfassung wird zwingend benötigt
- Messungen müssen über längere Zeit erfolgen

Ende

Vielen Dank

Vielen Dank für Eure Aufmerksamkeit!