

Software-Qualität

Kapitel 4

Messen von Software-Qualität

Inhalt
Software-Metriken
Metriken nach Maß: GQM

Prof. Dr. Kurt Schneider

Vage, konkret, messbar

Qualität (oder so)

K. Schneider / J. Greenyer

SWQ 2016 - 157

Überblick

- Was wird gemessen und wozu?
- In welchem Zusammenhang steht Messung?
 - Zusammenhang mit anderen Qualitäts-Maßnahmen
 - Voraussetzung für Planung und Kontrolle
 - Vorhersage von Fehlern, Risiken früh erkennen
 - Spezialfall Analytische Qualitätssicherung
- Metriken und Maße in der Software-Qualität
- „Metriken maßgeschneidert“: Goal-Question-Metric (GQM)

K. Schneider / J. Greenyer

SWQ 2016 - 158

Zusammenspiel der Maßnahmen

• Q-Management: Planen-lenken-beobachten-verbessern (Feedback)

Organisatorische Einbettung von Qualitätsaktivitäten

Messen

K. Schneider / J. Greenyer
Nach Martin Glinz: Vorlesung Software-Qualitätsmanagement, Universität Zürich, 1999

SWQ 2016 - 159

Was und woran misst man?

Beispiele

Produktmaße

Prozessmaße

Projektmaße

K. Schneider / J. Greenyer

SWQ 2016 - 160

Was wird gemessen?

Prinzip

Direkt	Indirekt am Produkt	Prozess
Codeumfang	Funktionalität	Produktivität
Laufzeit	Lesbarkeit	Nachvollziehbarkeit
Speicherbedarf	Wartbarkeit	Reife
Aufwand (PM)	Usability	Zertifizierbarkeit
Fehler	Komplexität	Agilität

aber wie misst man das?

K. Schneider / J. Greenyer

SWQ 2016 - 161

SE

Softwaremaße und -metriken

Definitionen

Messen

- Eigenschaften der realen Welt Zahlen oder Zeichen zuordnen

Maß

- Zuordnung einer Zahl oder Zeichens und einer Einheit (z.B. 1 m)

Metrik [gr.: „Kunst des Messens“) in der Mathematik

- Abstand zwischen zwei Dingen (axiomatisch definiert)

Softwaremetrik

Funktion, die eine Software-Einheit in einen Zahlenwert abbildet. Dieser Wert ist interpretierbar als der Erfüllungsgrad einer Qualitätseigenschaft der Software-Einheit. (IEEE Standard 1061)

SWQ
K. Schneider / J.Greenyer
SWQ 2016 - 162

SE

Messen und Skalen

Grundlagen

- Messen:** Abbilden eines (Software-) Objekts auf Skala
- Skala:** Mathem. Struktur aus
 - Grundmenge
 - Operationen
 - Relationen (Vergleiche)
- Aussagekraft der Metrik hängt von Skala ab
 - Welche Operationen sind darauf zulässig?
 - ACHTUNG:** meist weniger als auf Grundmenge definiert!
- Übler Fehler: Anwendung unzulässiger Operationen/Vgl.

z.B.

$\rightarrow (N, +, <, =)$

SWQ
K. Schneider / J.Greenyer
SWQ 2016 - 163

SE

Nominalskala

- Objekte werden von Metrik m „mit Namen versehen“ (nomen)
 - also klassifiziert
 - bzw. Schubladen zugeordnet
 - Es gibt nur eine Äquivalenzrelation \cong

Metrik m : „in Sprache geschrieben“

$\rightarrow \{ \text{Java, C, Scheme}, \cong \}$

SWQ

Erlaubt ist nur:

„Ist Programm A in gleicher Sprache geschrieben wie B?“

$m(A) \cong m(B) ?$

SWQ 2016 - 164

SE

Ordinalskala

- Objekte werden durch m klassifiziert
 - Es gibt eine Äquivalenzrelation \cong
 - Zusätzlich ist eine Ordnung definiert: „besser“ $<$

Metrik m : „wird benotet“

$\rightarrow \{ \{1,2,3,4,5\}, \cong, < \}$

SWQ

Erlaubt ist nun zusätzlich:

„Programm A wird besser benotet als B“

$m(A) < m(B)$

SWQ 2016 - 165

SE

Intervallskala

- Abstände (Intervalle) sind bedeutungsvoll
 - Intervall/Abstand zwischen aufeinanderfolgenden Werten ist „gleich groß“, z.B. $(3 \rightarrow 4 = 7 \rightarrow 8)$ oder $(1.5.11 \rightarrow 7.5.11 = 3.8.12 \rightarrow 9.8.12)$
 - Rechnen mit Intervallen ist erlaubt, mit Werten nicht $(7.5.11 \neq 7 + 1.5.11)$

Metrik m : „Termin“

$\rightarrow \{ \text{TageDifferenz}, +, -, \cdot, / \}$

SWQ

Erlaubt ist nun zusätzlich:

Intervalle ins Verhältnis setzen

Projekt ist *halb* so stark verspätet wie letztes

(d.h.: Intervall [t.Fertig-t.Geplant] ist *halb* so groß wie dort)

SWQ 2016 - 166

SE

Rationalskala

- Definierter Nullpunkt: man darf auch Verhältnisse bilden
 - Also zusätzlich multiplizieren, dividieren

Metrik m : „Anzahl Getter“

$\rightarrow (R^+, 0, \dots, \cdot, /)$

SWQ

Erlaubt ist nun zusätzlich:

„Programm A hat doppelt so viele Getter wie B“

Mittelwerte, Durchschnitte usw.

$m(A) = 2 \cdot m(B)$

SWQ 2016 - 167



Skalen-Beispiele um Software

- CMM Maturity Level (1-5)
- Beliebtheit einer App im AppStore (nach Downloads)
- Gefundene Fehler
- Zertifizierter Betrieb (ISO 9000)
- Bisher erreichte Leistungspunktzahl (Transcript of records)
- Wertung in Vorlesungsevaluierung

SWQ

K. Schneider / J.Greenyer

SWQ 2016 - 168



Klassische Metrik: Lines Of Code

Beispiel

LOC zählt die Codezeilen eines Programms.

LOC ist somit ein scheinbar einfaches Maß.

ABER:

Was wird gezählt? Und: was impliziert das?

- Nur Zeilen mit ausführbarem Code
- Ausführbarer Code und Datendefinition
- Ausführbarer Code, Definitionen und Kommentare
- Anzahl physikalischer Zeilen
- Begrenzung der Konstrukte zum Zählen verwenden

SWQ

K. Schneider / J.Greenyer

SWQ 2016 - 169

Klassische Metrik: Lines Of Code

Genauer hingesehen

Wie viele LOC hat dieses Programm?

```
public void buyTicket()
{
    // Adults pay more
    if ( isAdult )
    {
        // Payment delegated
        payFullFee();
    }
    else
    {
        // all others pay less
        payReducedFee();
    }

    //In all cases print the ticket
    printTicket();
}
```

alle Zeilen	17
ausführbarer Code	12
+Kommentare	16
Zeilen mit Semikolon	3

K. Schneider / J.Greenyer

SWQ 2016 - 170



Versuche, die Qualität zu messen

Bekanntestes Beispiel McCabe: Cyclomatic Complexity

Abgeleitet aus dem „Programmablaufgraph G“

$$V(G) = e - n + 2$$

edges, nodes

Algorithmus

gilt für strukt. Sprachen (Pascal, Cobol)

v(G) = Anzahl (IFs, Schleifen)

+ CASE-Fälle (je Anw. -1)

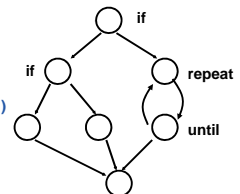
+ 1

Fehler-Risiko abschätzen

V(G) > 10 : mittel

V(G) > 20 : hoch

V(G) > 50 : unbeherrschbar



In diesem Beispiel:

9 edges – 7 nodes + 2

bzw.

2 ifs + 1 Schleife + 0 CASE + 1

= 4

SWQ

K. Schneider / J.Greenyer

SWQ 2016 - 171



Beispiel für McCabe-Berechnung

Auszug aus Beispielprogramm

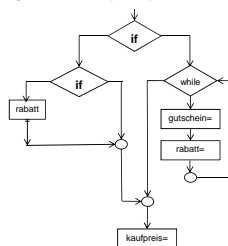
```
...
EuroWert kaufpreis=0;
EuroWert rabatt=0;
...
// Waren aussuchen, Kaufpreis ermitteln, dann:
if (kunde.istMitarbeiter()) {
    // Mitarbeitererrabatt, aber keine Gutscheine
    if (kunde.istRabattberechtigt()) {
        rabatt=kaufpreis*kunde.rabattSatz();
    }
}
else {
    // externer Kunde, hat evtl. Gutscheine
    while (kunde.hatGutschein()){
        gutschein = kunde.gibGutschein();
        rabatt = rabatt+gutschein.wert();
    }
}
kaufpreis=kaufpreis-rabatt;
...

```

Programmablaufgraphen ableiten:

Zwischenergebnis:

Programmablaufplan (DIN 66 001)



K. Schneider / J.Greenyer

SWQ 2016 - 172



Beispiel für McCabe-Berechnung

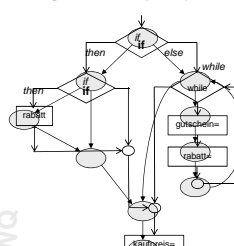
Programmablaufgraphen ableiten:

Zwischenergebnis:

Programmablaufplan (DIN 66 001)

Schritt 2 (Abstrahieren):

Programmablaufgraph



K. Schneider / J.Greenyer

SWQ 2016 - 173

SE Beispiel für McCabe-Berechnung

Programmablaufgraphen ableiten:
Zwischenergebnis:
Programmablaufplan (DIN 66 001)

Schritt 2 (Abstrahieren):
Programmablaufgraph

K. Schneider / J. Greenyer SWQ 2016 - 174

SE Beispiel für McCabe-Berechnung

Programmablaufgraph mit unnötigen Knoten

Schritt 3:
Reduzierter Programmablaufgraph ohne unnötige Knoten

Dies sind zwei gleichermaßen korrekte Programmablaufgraphen
 Beide mit $V(G1)=V(G2)=4$

K. Schneider / J. Greenyer SWQ 2016 - 175

SE Es gibt noch viel mehr Metriken

Beispiel: Metriken für OO

Traditionelle Maße für Objektorientierung modifiziert:
 Klassen, Objekte, Vererbung, Polymorphie als Basis

Objektorientierte Metriken nach Chidamber und Kemerer:

- **Weighted Methods per Class (WMC)**
 - McCabe für alle Methoden der Klasse, addiert.
- **Depth in Inheritance Tree (DIT)**
 - Wie viele Oberklassen darüber?
 - Je mehr, desto fehlerbehafteter.
- **Coupling Between Objects (CBO)**
 - Anz. Klassen, mit denen kommuniziert wird.
 - Je mehr, desto höhere Kopplung.

Diese Metriken werden auf Klassen angewendet.

K. Schneider / J. Greenyer SWQ 2016 - 176

SE Beispiele für Anforderungs-Metriken

- **Korrektheit und Verständlichkeit von Texten**
 - Fehlerfreiheit (Rechtschreibhilfe)
 - Anteil Passivsätze
 - Flesch-Kincaid und Flesch Reading Ease (Lesbarkeit)
 - Durchschnittliche Länge der Sätze
- **Sprachliche Defekte von Anforderungen**
- **Beispiel: Eindeutigkeit**

$$\text{Eindeutigkeit} = \frac{\sum \text{Anforderungen_Ohne_Defekte}}{\sum \text{Anforderungen}}$$
- **Beispiel: Testbarkeit**

$$\text{Testbarkeit} = \frac{\sum \text{Anforderungen_b_Zwei_Abnahmekriterien}}{\sum \text{Anforderungen}} \cdot \text{Eindeutigkeit}$$
 - Hat die Anforderung mind. zwei Abnahmekriterien?
 - Ist sie möglichst eindeutig formuliert?
 - Verweist sie auf die Abnahmekriterien?

M. Recknagel, C. Rupp: Metriken für Anforderungen. Wie gut sind Ihre Anforderungen wirklich? SQM, 11.5.06, Düsseldorf, SQS
 K. Schneider / J. Greenyer SWQ 2016 - 177

SE Kriterien für Projekt- und Produkterfolg

quasi ein Qualitätsmodell für die Spezifikation

Produktterfolg
 Fehler (Stichproben, Defect Management, des Testdirectors)
 Tests (Breitenprüfung)

Projekterfolg
 Featureumsetzung (Vergleich geplante-tatsächlich umgesetzte Features)
 Änderungshistorie
 Lieferantenentwicklungskosten
 Eigenentwicklungskosten
 Subjektive Erfahrungsberichte
 Gewährleistungs- und Kulanzkosten
 Aktionierungen
 Budget
 Zeiteinhaltung

M. Recknagel, C. Rupp: Metriken für Anforderungen. Wie gut sind Ihre Anforderungen wirklich? SQM, 11.5.06, Düsseldorf, SQS
 K. Schneider / J. Greenyer SWQ 2016 - 178

SE Fortschrittmessung mit Quality Gates

komplexe Metrik auf simpler Ordinalskala

• **Idee**

– **Kurze, scharfe Prüfung an definierten Prozess-Stellen**

Beispiel QGates

Bereit-für-Entwurf **Bf-Implementierung** **Bf-Auslieferung**

Prozess

– **Prüfkriterien: essenzielle Fortschrittsindikatoren**

- Dokumente vorhanden, zugänglich?
- Wichtige inhaltliche Prüfungen bestanden?

– **Fortschrittmessung: Passierte Quality Gates (hier: 1)**

• **Mehr dazu: bei Reviews (methodisch ähnlich)**

K. Schneider / J. Greenyer SWQ 2016 - 179

SE „Fallen“ bei Metriken

- Was misst man *wirklich*?
 - (Inwiefern) ist Metrik ein Modell für die gemessene Eigenschaft?
 - Bezug zu Mensch erfasst (z.B. Benutzbarkeit, Wartbarkeit)?
- Beispiel: „cyclomatic complexity is used as a
 - quantitative measure of *testability* [siehe bei Testen!] and an
 - indication of ultimate *reliability*“
- Was kann man aus den Ergebnissen schließen?
 - Auf welcher Skala liegen die Messungen?
- Generell: Vorsicht beim Interpretieren!
 - Gefährlicher Ansatz: „was können wir denn leicht messen?“
 - Ganz anderes Prinzip: Goal-Question-Metric

K. Schneider / J.Greenyer SWQ 2016 - 180

Q misst

Ziele
↓
Fragen
↓
Metriken

K. Schneider / J.Greenyer SWQ 2016 - 181

SE Wie findet man die richtigen Metriken?

- Situation:
 - Es gibt ein Problem
 - zu viele Fehler, zu viel Aufwand oder zu lange Entwicklungszeiten
 - Unternehmen oder Projekt möchte/muss sich verbessern
 - Dazu muss man etwas ändern
 - Aber was? Und wird dadurch wirklich etwas besser?
- Idee: Man müsste messen!
 - Nur was?
 - Die „normalen Metriken“ wie LoC, McCabe usw.?
 - Mit einem Wort: „was sich leicht messen lässt“?
- Das ist oft nicht die beste Lösung!
 - Geringe Aussagekraft für spezielles Problem
 - Oft fehlen in der Analyse wichtige Daten

Was tun?

K. Schneider / J.Greenyer SWQ 2016 - 182

SE Goal-Question-Metric (GQM)-Methode

Überblick

- Systematisches Vorgehen beim Messen von SW und Prozessen
- Top-Down: Ziele aufstellen, dazu passende Metriken ableiten
- GQM wurde erfolgreich in Industrieprojekten eingesetzt

K. Schneider / J.Greenyer SWQ 2016 - 183

SE GQM-Modell

K. Schneider / J.Greenyer SWQ 2016 - 184

SE Vorgehen bei GQM

Verfahren

Prinzip ist einfach:
Einige Ziele, mehr Fragen, möglichst wenige Maße

- Ziele erheben und verfeinern (Goal): Zielbaum
- Ziele mit Facetten genauer beschreiben (Goal)

Aspekte, um die es geht	- zum Beispiel beim Testen...
Zweck	Verstehen; verbessern
Qualitätsaspekt	Effizienz, Effektivität, Kostenwirksamkeit
Betrachtungsgegenstand	Testprozess, Testplan
Perspektive u. Umgebung	Projekt/Bereich xyz
- Ableitung von Fragen zu den Zielen (Questions): Ein Abstraction Sheet pro Ziel
- Ableitung von zugehörigen Metriken (Metric)
- Messplan für Datenerhebung erstellen (Metric)
- Datenerhebung und Auswertung

K. Schneider / J.Greenyer SWQ 2016 - 185

Messziele mit Facetten

Beispiel: Verständlichkeit des Codes

Ziel	Zweck	Q-Aspekt	Beobachtungsgegenstand	Perspektive
3.1	Untersuche	Lesbarkeit	Kommentare im Code	Entwickler
3.2	Verbessere	Lesbarkeit	Kommentare im Code	Tester
...
5.1	Steuere	Effizienz	Ablauf Modultest	Projektleitung
...

Facetten führen zu Nachfragen, Umformulierung hier als Spalten notiert

K. Schneider / J.Greenyer SWQ 2016 - 186

Abstraction Sheet

Beispiel

Ziel: G 3.2 Abstraction Sheet Ausgefüllt von: Q Datum: 14.2.

Zweck der Messung	Qualitätsaspekt	Betrachtungsgegenstand	Perspektive
Verbessere	Lesbarkeit	Kommentare im Code	Tester

Qualitätsfaktoren

a- Kommentardichte
b- sprachlich verständlich
c- Bezug zum Anwendungsglossar
d- mit Begründungen (Rationale)

Einflussfaktoren

- Forderungen in Programmierrichtlinien
- Englischfähigkeiten
- Schulung
- Moderierter Erfahrungsworkshop zum Kommentarstil

Ausgangshypothese: wie ist es jetzt?

a- unter 5% der Zeilen sind Kommentare
b1- ca.70% enthalten nur Stichwörter, aber keine vollständigen Sätze
b2- schlechtes Englisch
c- keine Referenzen auf Glossar (<1%)
d- ca. % der Kommentare beziehen sich darauf, wie es funktioniert – nicht, wieso es so gemacht wird

Einflusshypothese: Abhängigkeiten

- Forderungen in Programmierrichtlinien beeinflussen (a) und (b) positiv
- an den Englischfähigkeiten lässt sich kurzfristig nichts ändern (b2)
- Durch Schulung können Entwickler lernen, Glossar zu nutzen (c)
- Moderierter Erfahrungsworkshop zu gutem Kommentarstil wirkt sich auf alle positiv aus, auch (d)

K. Schneider / J.Greenyer SWQ 2016 - 187

Abstraction Sheet

Zusammenhang

Messen

Ziel: G 3.2 Abstraction Sheet Ausgefüllt von: Q Datum: 14.2.

Zweck der Messung	Qualitätsaspekt	Betrachtungsgegenstand	Perspektive
Verbessere	Lesbarkeit	Kommentare im Code	Tester

Qualitätsfaktoren

a- Kommentardichte
b- sprachlich verständlich
c- Bezug zum Anwendungsglossar
d- mit Begründungen (Rationale)

Einflussfaktoren

- Forderungen in Programmierrichtlinien
- Englischfähigkeiten
- Schulung
- Moderierter Erfahrungsworkshop zum Kommentarstil

Ausgangshypothese: wie ist es jetzt?

a- unter 5% der Zeilen sind Kommentare
b1- ca.70% enthalten nur Stichwörter, aber keine vollständigen Sätze
b2- schlechtes Englisch
c- keine Referenzen auf Glossar (<1%)
d- ca. % der Kommentare beziehen sich darauf, wie es funktioniert – nicht, wieso es so gemacht wird

Einflusshypothese: Abhängigkeiten

- Forderungen in Programmierrichtlinien beeinflussen (a) und (b) positiv
- an den Englischfähigkeiten lässt sich kurzfristig nichts ändern (b2)
- Durch Schulung können Entwickler lernen, Glossar zu nutzen (c)
- Moderierter Erfahrungsworkshop zu gutem Kommentarstil wirkt sich auf alle positiv aus, auch (d)

Zustand Einflüsse und Abhängigkeiten

K. Schneider / J.Greenyer SWQ 2016 - 188

Ableitung einer Fragestellung und Metrik

Beispiel

- Ableitung von Fragestellungen aus dem Abstraction Sheet sowie zugehörige Metriken

Question: Führt ein *moderierter Erfahrungsworkshop zum Kommentarstil* dazu, dass Entwickler auch Begründungen (Rationale) in den Kommentaren dokumentieren?

Metric: *Anteil der neuen Kommentare, die eine Begründung enthalten*. Vor und nach der Durchführung eines Erfahrungsworkshops, zum Vergleich.

K. Schneider / J.Greenyer SWQ 2016 - 189

Auswertung der Messungen

- Messungen werden in einer **Feedback Session** diskutiert
 - Beispiel: Nach Durchführung des Erfahrungsworkshops enthalten 25% der neuen Kommentare *keine* Begründung (Rationale). Ausgangshypothese war 75%. Ist das eine *Verbesserung der Lesbarkeit von Kommentaren*?
- Rückschluss aus dem Ergebnis der **Feedback Session** auf Fragestellung und jeweiliges Ziel (Goal)
- Dokumentierte Ergebnisse der Auswertungsphase
 - Beobachtungen
 - Interpretierte Messergebnisse
 - Schlussfolgerungen
 - Erforderliche Aktivitäten

Planung Messung

Definition

Goal Question

Metric

Zielerreichung

Beurteilung

Interpretation

Datensammlung

K. Schneider / J.Greenyer SWQ 2016 - 190

Zusammenfassung GQM

Messen, was man wissen will, nicht das, was leicht zu messen ist!

- Mit GQM findet man systematisch geeignete Metriken (vom Ziel zur Metrik). Manchmal denkt man sich neue aus.
- Weglassen ist die Kunst: Wenige prägnante Fragen, wenige Metriken
- Das Ergebnis ist dann leicht interpretierbar: Einsetzen ins Abstraction Sheet, mit Erwartung vergleichen
- Die Messergebnisse sind selten statistisch signifikant, aber sehr häufig aussagekräftig und nützlich

K. Schneider / J.Greenyer SWQ 2016 - 191