

Agenda: Metriken

- **Programmierrichtlinien**
- **Software-Metriken**
- **Software-Metriken: Praxis**
- **Metriken im Qualitätsmanagement**
- **McCabe-Metriken**
- **Metriken zur Planung, Kontrolle und Steuerung des Software-Tests**
- **Zusammenfassung Metriken**

Aufgabe

Überlegen Sie sich *Programmierrichtlinien* als QS Maßnahmen

Zusammenarbeit mit Nachbarn erwünscht

Bitte, aufschreiben mit Erklärung und Ziel
(Motivation, was bringt's)

Zeit: 5 min.

Anschließendes Sammeln und Besprechen

Programmierrichtlinien (1)

- Verwende wenige Standard-Steuerkonstrukte
- Nicht zu tief schachteln (max. 4 Stufen)
- Keine GOTO's
- Anwenderbezogene Datentypen (eindeutige Namen)
- Geheimnisprinzip, Datenkapselung
- Maschinenabhängigkeiten isolieren (Subroutine)
- Begrenzung der Anzahl der Anweisungen (25) und Parameter (5) einer Subroutine

Programmierrichtlinien (2)

- Markierung verwenden (Einrücken, Klammern, Leerzeilen, Umrandungen)
- Übersichtliche Codes verwenden (keine übertriebene Minimierung der Anzahl Anweisungen)
- Subroutinen eindeutige Funktionen zuweisen
- Keine Optimierung ohne Abschätzung des Nutzens
- Variablen nicht mehrfach verwenden
- Auf richtige Initialisierung achten (Plausi-Prüfung)

Programmierrichtlinien 3

- Ein- Ausgaberaum begrenzen (Grenzwertprüfung, Ende durch Markierungszeichen)
- Testpunkte einsetzen (Zwischenergebnisse, Zähler evtl. in Ringpuffer oder auf Datei schreiben)
- Kommentare knapp und präzise (Daten: bei deren Deklaration)
- Programm-Module mit eindeutigem Verhalten (Modul hat kein Erinnerungsvermögen)

Hinweis: *Programmierrichtlinien können auch als Metriken genutzt werden, wobei diese zur Überprüfung der Einhaltung herangezogen werden (sogar maschinell).*

Software-Metriken

- Software-Metriken sind als quantitative Methoden zur Verbesserung der Software-Qualität und -produktivität zu verstehen.
- Zu Beginn steht ein dokumentierter Entwicklungsprozess, der schrittweise verbessert wird gemäß der gestellten Zielvorgaben, die auf Unternehmensleitsätzen beruhen.
- Zur Messung der periodischen Fortschritte sind Metriken zu definieren.
- Die gemessenen Daten sind einerseits Bestätigungsinformation, andererseits aber auch Warnsignale für Problembereiche im Entwicklungsprozess, die Verbesserungsmaßnahmen hervorrufen. Hier handelt es sich um den Deming-Kreis mit Plan-Do-Check-Act (PDCA).

Software-Metriken

Metriken spielen die Schlüsselrolle bei diesem Rückkopplungsprozess. Die erreichten Prozessverbesserungen führen zu Produkten höherer Qualität. Eine höhere Produktivität des Entwicklerteams stärkt die Konkurrenzfähigkeit.

Ein Programm zur Einführung von Metriken hat viele Vorteile:

- Niedrigere Entwicklungskosten wegen höherer Produktivität
- Höherer Kundenzufriedenheit infolge verbesserter Produktqualität
- Niedrigere Kosten, höhere Verkaufszahlen, höheren Gewinne

Wie bei TQM auch, muss hinter der Einführung von Software-Metriken die Geschäfts- bzw. Bereichsleitung stehen.

* Quelle: AkSeQM

Software-Metriken

- Der Gebrauch von Metriken verbessert die Fähigkeit der Organisation, neue Projekte zu planen und zu realisieren.
- Wenn historische Projektdaten vorliegen, können zwischen neuen und vorhergehenden ähnlichen Projekten Vergleiche angestellt werden.
- Es können Kosten und Terminpläne neuer Projekte besser eingeschätzt und entschieden werden.
- Anhand der zur Verfügung stehenden Messdaten ist es den Projektleitern möglich, Projektpläne besser zu planen und zu korrigieren, was einen erfolgreichen Abschluss des Projektes erhöht.
- Die Messdaten helfen dem Projektleiter bei der Bewertung seiner organisatorischen oder technischen Maßnahmen und deren Alternativen und dienen zur Beurteilung von Projektstatus, Qualitätsniveau der Produkte und zur Produktfreigabe.

* Quelle: AkSeQM

Software-Metriken

Die Einführung eines Metrikprogrammes führt zu wachsendem Kundenvertrauen, weil es zeigt,

- das Unternehmen die Stärken und Schwächen seiner Entwicklungs- und Geschäftsprozesse und Produkte kennt,
- das Unternehmen Steuerungsinstrumente zum Erkennen von Prozessschwächen und Qualitätsveränderungen der Produkte nutzen
- das Unternehmen Maßnahmen zur Beseitigung von Schwächen auf der Prozessebene und an den Produkten selbst ergreift

* Quelle: AkSeQM

Software-Metriken

- Ein Metrikenprogramm verhilft dem gesamten Entwicklungsprozess zu mehr Transparenz und verringert so dessen scheinbare Komplexität.
- Der Einsatz von Metriken ist ein wichtiger Teil eines ganzheitlichen Ansatzes beim fortwährenden Suchen nach Qualitäts- und Produktivitätsverbesserungen zum Erlangen einer unternehmensweiten Qualitätskultur.
- Durch ihre Angleichung an andere industrielle Entwicklungsprozesse wird die Softwareentwicklung verständlicher.
- Die bloße Einführung eines Metrikenprogrammes führt alleine aber noch nicht zu all diesen Verbesserungen, gestattet aber, Änderungen des Entwicklungsprozesses zu bewerten.

* Quelle: AkSeQM

Software-Metriken

- Es hat sich früh gezeigt, dass der Einsatz von Softwaremetriken eine effektive Technik zur Steigerung der Qualität und Produktivität von Software ist.
- Der Einsatz begann in den 80er Jahren. Die Grundlagenarbeit wurde in den 70er Jahren geleistet.
- Heutige Einsätze von Softwaremetriken verwenden globale Indikatoren, die Einblicke in die Verbesserungspotentiale von Entwicklungs- und Wartungsprozessen erlauben. Die Qualitäts- und Produktivitätsziele des Unternehmens werden durch Verbesserungen am Entwicklungsprozess erreicht. Tendenziell ist zu erkennen, dass Metriken mehr und mehr zur Fehlervermeidung eingesetzt werden und nicht nur zur Fehleranalyse innerhalb eines Rückmelde-mechanismus.

* Quelle: AkSeQM

Software-Metriken

Für Software-Metriken kann es sechs Einsatzfelder geben:

- Überwachung von quantitativen Zielvereinbarungen
- Verbesserung der Qualität in Verbindung mit einem unternehmensweiten Programm wie z. B. TQM
- Steigerung der Produktivität mit eigenen Zielen und gesonderten Maßnahmen
- Projektplanung - Einschätzung der Komplexität, der Termine, des Personalbedarfs und des Budgets
- Entwicklungsmanagement - Steuerung und Überwachung von Softwareprojekten
- Erhöhung der Kundenzufriedenheit durch Bekanntmachung der Bemühungen um Verbesserungen von Produktivität und Qualität (Zertifizierung, geprüft nach ...)

* Quelle: AkSeQM

Software-Metriken

Die Einführung eines Metrikprogrammes unterteilt sich in folgende Schritte:

- Entwicklungsprozess definieren und dokumentieren, damit die Folge von Phasen nachvollziehbar ist und verstanden wird
- Ziele setzen
- Verantwortliche benennen. Benötigte Eigenschaften des Leiters eines Metrikprogrammes sind z. B. Motivation, Kundenorientierung, gute kommunikative Fähigkeiten, gute Planung, diplomatisches Geschick, Glaubwürdigkeit, Liebe zum Detail und Genauigkeit, organisatorische Fähigkeiten, Ausdauer und Beharrlichkeit sowie Macht und Autorität.
- Vorarbeiten durchführen wie (firmeninterne) Kundenbefragungen, Sammlung von Ausgangsdaten durch überprüfen jetziger Praktiken.

* Quelle: AkSeQM

Software-Metriken

Metriken definieren:

- objektive Metriken wie
 - Programmgröße
 - Lines of Code
 - Function Points
 - Aufwand
 - Terminplan
 - Anzahl der Fehler
 - Qualitätskosten
- subjektive Metriken
 - schwer quantifizierbare Sachverhalte wie die Kundenzufriedenheit und Benutzerakzeptanz

* Quelle: AkSeQM

Software-Metriken

- Für die strategische Planung Zusammenfassung mehrere Phasen des Software-Entwicklungsprozesses, um mit den vorgegebenen Metriken dem Management einen Überblick über Produktivität und Qualität zu geben
- Entsprechende Phasenmetriken für die taktische Planung der einzelnen Phasen:
 - Metriken der Codeinspektion sind z. B. Phasenmetriken der Codierungsphase
 - Festlegung von notwendigen Zusatzmetriken
- Verkaufsstrategie für das Metrikprogramm entwickeln, um die Akzeptanz und Unterstützung von allen Beteiligten zu erreichen
- Rückmeldungen auswerten und mit den Zielen vergleichen

* Quelle: AkSeQM

Software-Metriken

Die Durchführung eines Metrikprogrammes kann auf fehlende Akzeptanz treffen, wobei folgende Argumenten aufgeführt werden:

- Metriken schränken die Kreativität ein
- Metriken verursachen zusätzliche Arbeit
- Der Nutzen bleibt unklar
- Angst vor persönlichen Leistungsmessungen.

Vor unrealistischen Erwartungen sei gewarnt. Nur schrittweise führt ein Metrikprogramm zu Verbesserungen der Produktivität und Qualität eines Software-Entwicklungsprozesses.

Einen gewissen Einfluss auf den Erfolg eines Metrikprogrammes haben auch Werkzeuge zur Erfassung der Metriken und der Unterstützung des Berichtswesens.

* Quelle: AkSeQM

Software-Metriken

Die Durchführung eines Metrikprogrammes kann auf fehlende Akzeptanz treffen, wobei folgende Argumenten aufgeführt werden:

- Metriken schränken die Kreativität ein
- Metriken verursachen zusätzliche Arbeit
- Der Nutzen bleibt unklar
- Angst vor persönlichen Leistungsmessungen.

Vor unrealistischen Erwartungen sei gewarnt. Nur schrittweise führt ein Metrikprogramm zu Verbesserungen der Produktivität und Qualität eines Software-Entwicklungsprozesses.

Einen gewissen Einfluss auf den Erfolg eines Metrikprogrammes haben auch Werkzeuge zur Erfassung der Metriken und der Unterstützung des Berichtswesens.

* Quelle: AkSeQM

Software-Metriken: Die Praxis

Objektive Maße zum Verständnis, zur Beurteilung und zu Kontrolle von Software-Entwicklungsprozesse sind:

- Versagenshäufigkeit
- Fehleranzahl und Änderungshäufigkeit, Fehlerentdeckungsrate
- Umfangsmetriken
- Testabdeckungsmetriken
- Zeit- und Aufwandsmetriken
- Strukturmetriken

* Quelle: AkSeQM

Software-Metriken: Die Praxis

Erstrebenswerte und erreichbare Ziele sind:

- verbesserte Kommunikationskultur
- verbesserte Produktqualität
- erhöhte Produktivität im Entwicklungsprozess
- objektiv steuerbare Prozesse

* Quelle: AkSeQM

Software-Metriken: Die Praxis

Wichtigste Erfolgsfaktoren dafür:

- Metriken grundsätzlich prozess- und produktorientiert (niemals personenbezogen)
- Metriken pro-aktiv, integriert im Tagesgeschäft, weil diese dann ernst genommen werden
- Einfache Metriken, in Ziel und Auswirkungen nachvollziehbar. Komplexe Metriken und andere aufwendige Modelle sind bei der Einführung nicht empfehlenswert
- Sammlung in eigenen Projekten und nach Möglichkeit nicht durch externe Planer oder Metrikgruppen
- Angemessener Trainingsaufwand und genügend exakte Bereitstellung von Anleitungen und Verfahren
- Unterstützung durch das Management

* Quelle: AkSeQM

Software-Metriken: Die Praxis

Eigenschaften von Software sind:

- Aus (unsichtbaren) Teilen zusammengesetzte Aggregate, dessen Qualität von der Güte und der Dokumentation der logischen Strukturierung dieser Einzelteile abhängt
- Kontinuierlich sich weiterentwickelnde Anforderungen, was eine gewisse Anpassbarkeit bedingt
- Einzigartigkeit jedes Produktes, da es individuell entwickelt und nicht als Kopie produziert wurde
- SW-Entwicklung und –Qualität hängt von den beteiligten Menschen, deren Ausbildung, Qualifikation, Erfahrung und auch deren Motivation ab
- Eigenschaften und Entwicklungsablauf sind schwer einschätz- und vorhersagbar, da entsprechende Modelle zur Analyse und Verbesserung fehlen

Es ist erkennbar, dass unter TQM verstandene Ansätze nicht ohne weiteres auf die Software-Entwicklung anwendbar ist.

* Quelle: AkSeQM

Software-Metriken: Die Praxis

Als Vorgehensweisen, um etwa das QIP (Quality Improvement Paradigm) zu entwickeln gibt es folgende Ansätze:

- die ingenieurwissenschaftliche,
- die empirische,
- die mathematischen.

* Quelle: AkSeQM

Das QIP (Quality Improvement Paradigm) erfolgt in sechs Schritten

1. Charakterisierung des Projektes und Identifikation existierenden Know-how's
2. Definition der Ziele des Projektes in messbarer Form
3. Wahl geeigneter Prozesse und Zusammenstellung des Projektplanes
4. Durchführung des Projektes gemäß Plan-Erfassung und Verwendung von Daten zur Projektkontrolle
5. Post-mortem-Analyse des Projektverlaufes, Verbesserungsvorschläge
6. Dauerhafte Sicherung der Erfahrungen

* Quelle: AkSeQM

Software-Metriken: Die Praxis

Analog zum QIP-Ansatz ist das zielorientiertes Messen mit dem GQM-Ansatz (Goal Question Metric) nach Basili, Rombach (Universität Kaiserslautern) wie folgt strukturiert:

- Charakterisiere
 - Identifiziere GQM-Ziele
- Entwickle GQM-Pläne
 - Entwicklung Messplan
 - Sammlung und Validierung der Daten
- Analysiere und interpretiere Daten projektbegleitend
 - Analysiere und interpretiere post mortem
 - Sichere Ergebnisse

* Quelle: AkSeQM

Software-Metriken: Die Praxis

Ergebnisse sind in einem zentralen GQM-Plan als zentrales Dokument zusammen zu führen. Techniken zur Darstellung werden auf folgenden Ebenen angewandt:

- konzeptuellen Ebene
- operativen Ebene
- quantitativen Ebene

Als Werkzeug gibt es einen grafischen GQM-Plan-Editor (GQM-DIVA), basierend auf einer formalen Sprache.

Weitere Ansätze zum zielorientierten Messen sind der QFD- und SQM-Ansatz.

* Quelle: AkSeQM

Software-Metriken: Die Praxis

Dem **QFD-Ansatz** (Quality Function Deployment) als ISO9001-konforme Qualitätsmethode liegt das Prinzip zugrunde, dass Charakteristika des zu erzeugenden Endprodukts sich auf verwandte Charakteristika von Zwischenprodukten wie z. B. Anforderungen oder Entwurf beziehen. Der QFD-Ansatz stellt aber keine Anleitung für die Ableitung von interessierenden Charakteristika und Maßen aus den Zielen bereit.

Beim **SQM-Ansatz** (Software Quality Metrics) konzentrieren sich die Ziele der Messungen auf das Endprodukt aus der Sicht des Kunden oder Anwenders mit dem Zweck der Bewertung und Zertifizierung. Dabei beschränkt sich die Anwendung auf eine vordefinierte Hierarchie derjenigen Qualitätsfaktoren, die für die Zielgruppe wichtig sind.

* Quelle: AkSeQM

Software-Metriken: Die Praxis

Im Gegensatz zu **QFD**- und **SQM**-Ansatz können beim **GQM**-Ansatz beliebige Ziele zu beliebigen Zwecken und aus beliebigen Blickwinkeln definiert werden, wobei unter Betrachtung des Kontextes und mit flexiblen Mechanismen Ableitung von Maßen für Produkte oder auch Prozesse möglich sind.

Das explizite Modellieren von Software-Produkten und Prozessen ist mittels einer Prozeßmodellierungssprache, wie MVP-L (Multi View Process Modeling Language), entwickelt an der Uni Kaiserslautern, möglich.

Umfassende Wiederverwendung, nicht nur von Code, sondern auch Zwischenprodukten wie Spezifikationen, Entwurf und auch Entwicklungsprozessen sind anzustreben.

* Quelle: AkSeQM

Software-Metriken: Die Praxis

Qualitätsverbesserungs-Programme nach dem QIP (Quality Improvement Paradigm) beginnen mit einem Pilotprojekt mit folgenden Aspekten:

- Entlangmanövrieren des Programms an auf konkret messbaren Ebenen hinunter verfeinerten Verbesserungszielen
- Kleine überschaubare Schritte (Prinzip der kleinen Schritte)
- Einbeziehung aller tangierten Mitarbeiter
- Hauptziel ist die Verbesserung der Produkte
- Zusicherung der Unterstützung des (Top-)Managements
- Erzielen und verbreiten früher Erfolge (Quick wins)
- Eigene Verbesserung ist das Ziel und nicht Vergleiche mit anderen
- Änderungen frühzeitig (und verträglich) ankündigen
- Eine Experience Factory frühzeitig einrichten
- Das Programm rechtzeitig erweitern und ausweiten (grown up)

* Quelle: AkSeQM

Metriken im Qualitätsmanagement

Software-Qualität wird definiert als die Gesamtheit von Eigenschaften und Merkmalen eines Software-Produkts, die geeignet sind, vorausgesetzte und vereinbarte Erfordernisse dieses Software-Produktes zu erfüllen, entsprechend DIN 8402.

Ein Qualitätsmanagement ist demnach die Summe der geplanten systematischen Tätigkeiten, die Qualität schaffen und kontrollieren.

Software-Qualitätsmetriken umfassen jene qualitativen Messverfahren, die die Erreichung von Qualitätssicherungszielen in Prozessen, an Produkten oder bei Ressourcen validieren oder bewerten.

* Quelle: AkSeQM

Metriken im Qualitätsmanagement

Unter Software-Qualitätsmetriken fallen folgende Verfahren:

- Reviews und Inspektionen (z. B. mittels Checklisten)
- Komplexitätsmetriken (z. B. Kontrollfluss mit Verschachtelung)
- Testmetriken (z. B. Lesbarkeit)
- Verfolgung von (Kunden-)Anforderungen, Entwurfsentscheidungen
- dynamische Testverfahren (z. B. Testabdeckung)
- Datenanalyse (z. B. Datenbankzugriffe)
- Kostenmodelle (z. B. Kosten-Nutzen-Rechnung)
- Risikoabschätzung (z. B. Priorisierung identifizierbarer Risiken)
- Ausführungszeitmodellierung (z. B. Antwort- und Lastverhalten)
- statistische Modelle (z. B. Operational Profiles)
- Prozessbewertung (z. B. Effektivität von Inspektionen)
- Ressourcenabschätzung (z. B. Effizienz von Modultests)

* Quelle: AkSeQM

Metriken im Qualitätsmanagement

Prinzipiell lassen sich für alle Qualitätskriterien Ziele und passende Metriken vereinbaren. Freilich sind manche dieser Qualitätsmetriken erst sehr spät oder nur subjektiv erfaßbar (Kundenzufriedenheit z. B:).

Deshalb wird zwischen direkten und indirekten Qualitätsmetriken unterschieden.

* Quelle: AkSeQM

Metriken im Qualitätsmanagement

Typische direkte Metriken sind

- Metriken für Qualitätsfaktoren wie Wartbarkeit oder Verständlichkeit
- produktorientierte Qualitätsmetriken (Fehlerdichte, MTTF, Rate der Kundenfehlermeldungen, Kundenzufriedenheit, Laststabilität),
- prozessorientierte Qualitätsmetriken (phasenorientierte Fehlerentdeckungsraten, Korrekturfehler, Fehlerdichte nach Modultest, Testabdeckungsrate),
- wartungsorientierte Qualitätsmetriken (Korrekturaufwand, Korrekturzeit, Anzahl von patches, Fehlerrate bei Korrekturen, Verzögerungszeit der Korrekturen nachdem die Fehler gemeldet wurde).

* Quelle: AkSeQM

Metriken im Qualitätsmanagement

Grundsätzlich sind direkte Qualitätsmetriken gegen Ende des Entwicklungsprozesses oder im Feld zuverlässiger und stabiler als in den frühen Phasen der Entwicklung.

Die aus der Software direkt messbaren Aspekte, wie z. B. die Komplexität, sind dagegen indirekte Qualitätsmetriken, denn eine Aussage über die Qualität ist nur über Erfahrungen aus realen Projekten möglich. Daher werden solche indirekten Metriken auch als Indikatoren bezeichnet.

* Quelle: AkSeQM

Metriken im Qualitätsmanagement

Grundsätzlich sind direkte Qualitätsmetriken gegen Ende des Entwicklungsprozesses oder im Feld zuverlässiger und stabiler als in den frühen Phasen der Entwicklung.

Die aus der Software direkt messbaren Aspekte, wie z. B. die Komplexität, sind dagegen indirekte Qualitätsmetriken, denn eine Aussage über die Qualität ist nur über Erfahrungen aus realen Projekten möglich. Daher werden solche indirekten Metriken auch als Indikatoren bezeichnet.

Beziehungen zwischen direkten Qualitätsmetriken und indirekten - aber direkt aus der Software messbaren - Qualitätsmetriken werden durch empirisch erhobene Qualitätsmodelle festgelegt.

Dabei sind die genannten Komplexitätsmetriken der analytischen Qualitätssicherung zuzuordnen, da sie nur den Ist-Zustand feststellen und selbst nicht konstruktiv sind.

* Quelle: AkSeQM

McCabe-Metriken

Im Bereich quantitativer Verfahren der Software-Entwicklung konzentrierte sich die Aufmerksamkeit auf Komplexitäts-metriken als Qualitätsindikatoren, insbesondere der von McCabe.

Aspekte der Software-Qualität:

- Testbarkeit
- Testabdeckung
- Wartbarkeit
- Redundanz
- Wiederverwendbarkeit
- Komplexität
- Modularität

* Quelle: AkSeQM

McCabe-Metriken

Warum wird Qualität gemessen ?

- Verbesserung der Qualität
- Planung von Kosten und Ressourcen
- Wartung
- Test

Software-Metriken beschreiben

- Wartbarkeit
- Zuverlässigkeit
- Kosten
- Zeitpläne

* Quelle: AkSeQM

McCabe-Metriken

Metriken

- McCabe-Metriken
- Halstead-Metriken
- Line Count Metriken
- Modified Condition / Decision Metriken

Integration Level Metriken

- Design-Komplexität
- Integration-Komplexität
- Global Data Komplexität

* Quelle: AkSeQM

McCabe-Metriken

Unit Level Metriken

- Zyklomatische Komplexität
- Essentielle Komplexität
- Modul Design Komplexität
- Specified data Komplexität
- Lines of Code
- Halstead-Metriken

* Quelle: AkSeQM

McCabe-Metriken

Objektorientierte Metriken

- Polymorphismen
- Anzahl Wurzeln
- Prozentsatz der "overloaded calls"
- Vererbung
- Tiefe
- "Fan-in"
- Anzahl Kinder
- Verkapselung
- Prozentsatz an "public data"
- Zugriff auf "public data"

* Quelle: AkSeQM

McCabe-Metriken

McCabe's Zyklomatische Komplexität:

$v(G)$ = Anzahl der linear unabhängigen Pfade durch ein Modul

McCabe's Essentielle Komplexität:

$ev(G)$ = Maßzahl für den Anteil unstrukturierter Konstrukte in einem Modul

McCabe's Integrations-Komplexität:

$iv(G)$ = Maßzahl für den Integrationstest-Aufwand

Warum Komplexitätsmessung ?

- Aufwand des Verstehens ist proportional zur Komplexität des Codes
- Testaufwand sollte proportional zur Komplexität des Codes sein

* Quelle: AkSeQM

McCabe-Metriken

Software niedriger Komplexität

- Zuverlässig
- Einfache Logik
- Nicht fehleranfällig
- Leicht zu testen
- Wartbar
- Gut strukturiert
- Leicht zu verstehen
- Leicht zu modifizieren

* Quelle: AkSeQM

McCabe-Metriken

Software mittlerer Komplexität

- Nicht sehr zuverlässig
- Teils komplizierte Logik
- Fehleranfällig
- Schwierig zu testen
- Wartbar
- Kann verstanden werden
- Kann modifiziert werden
- Komplexität kann reduziert werden

* Quelle: AkSeQM

McCabe-Metriken

Software hoher Komplexität

- Unzuverlässig
- Fehleranfällig
- Sehr schwierig zu testen
- Nicht oder nur mit sehr hohem Aufwand wartbar
- Schwierig zu verstehen
- Kaum zu modifizieren
- Reduzierung von Komplexität kaum möglich

* Quelle: AkSeQM

Metriken zur Planung, Kontrolle und Steuerung des Software-Tests

Der Software-Test ist eine nie perfekt durchführbare Aktivität. Daher ist es allein mit Metriken möglich, die gewünschte Qualität des Software-Tests und damit die geforderte Qualität des Softwareproduktes zu gewährleisten.

Metriken als wichtige Hilfsmittel für folgende Aspekte des Software-Tests:

- Testplanung
- Testkontrolle
- Teststeuerung

* Quelle: AkSeQM

Metriken zur Planung, Kontrolle und Steuerung des Software-Tests

Testplanung

Auswahl der Testverfahren gestützt auf Metrikwerte

- Fehlerstatistiken
- Komplexitätsmetriken
- Kontrollflußmetriken (s. McCabe)
- Datendeklarationsmetriken
- Datenflußmetriken
- Arithmetikmetriken

* Quelle: AkSeQM

Metriken zur Planung, Kontrolle und Steuerung des Software-Tests

Testkontrolle

Testkontrolle des Testfortschritts und der Erreichung von Testzielen mittels Metriken

Teststeuerung

Problemidentifikation und steuernder Eingriff anhand von Metrikwerten

Für Testkontrolle und Teststeuerung gilt:

- Funktionsorientierter Test
- Strukturorientierter Test
- Kontrollflussorientierte Testverfahren
- Testoptimierung

* Quelle: AkSeQM

Metriken zur Planung, Kontrolle und Steuerung des Software-Tests

- Law of Demeter
- Metriken Chidamber und Kemerer
- Metrik WMC: Gewichtete Menge an Methoden (Weighted methods per class)
- Metrik DIT: Tiefe im Hierarchiebaum (Depth of inheritance tree)
- Metrik NOC: Anzahl der direkten Unterklassen (Number of children)
- Metrik CBO: Kopplung zwischen Klassen (Coupling between classes)
- Metrik RFC: Menge aller genutzten Methoden (Response for a class)
- Metrik LCOM: Kohäsionsmangel der Methoden (Lack of cohesion in methods)
- Metriken von Sharble und Cohen
- Metrik WAC: Gewichtete Menge an Variablen (Weighted attributes per class)
- Metrik NOT: Anzahl der Methodenparameter (Number of tramps)
- Metrik VOD: Verstöße gegen das Law of Demeter (Violations of the Law of Demeter)

* Quelle: AkSeQM

Zusammenfassung Metriken

- Software-Komplexitätsmetriken werden direkt anhand des Codes kalkuliert
- Sie sind unabhängig von der Programmiersprache
- Sie sind präzise und objektiv
- Sie unterstützen bei der Abschätzung von Zuverlässigkeit und Wartbarkeit
- Sie unterstützen beim Testen und Reengineering

* Quelle: AkSeQM