

## HIS Source Code Metriken

*Stand:* 01.04.2008  
*Version:* 1.3.1  
*Autor:* Kuder, Helmar

*Mitarbeit:*

Audi  
BMW Group  
Daimler

Porsche  
Volkswagen

Albrecht Korn, Erwin Haunschild  
Dr. Bernhard Kalusche  
Martin Huber, Andreas Krüger,  
Dr. Eric Sax (MBtech), Helmar Kuder (MBtech)  
Dr. Rüdiger Dorn, Jesper Hansson  
Herbert Tschinkel, Jörg Kluge

## Inhaltverzeichnis

<b>1</b>	<b>GRUNDLAGEN.....</b>	<b>3</b>
1.1	Allgemein .....	3
1.2	Referenzierte Dokumente .....	3
1.3	Glossar .....	3
<b>2</b>	<b>UMSETZUNG.....</b>	<b>4</b>
2.1	Erhebung der Metriken .....	4
2.2	Anwendung der Metriken .....	4
<b>3</b>	<b>METRIKEN – MAßZAHLEN .....</b>	<b>5</b>
3.1	Metriken mit Grenzwerten .....	5
3.2	Metriken ohne Grenzwerte.....	8

## Historie

Version	Datum	Veränderungen
1.0	31.05.2005	Erste offizielle Version
1.1	30.11.2005	Entfernung Referenz auf Dokument „HIS Anforderungen an den Softwaretest“ Neues Glossar
1.2	14.02.2006	Aktualisierung Tabelle 3-1 und 3-2, Rechtschreibung, Grenzwert ap_cg_cycle, Verbindlichkeit Metriken ohne Grenzwerte
1.3	11.05.2007	Aktualisierung der Kommentare zu den Metriken CALLING, Stabilitätsindex $S_i$ , NOMV, NOMVPR und ap_cg_cycle.
1.3.1	01.04.2008	Erweiterung Titel für ap_cg_cycle: "Anzahl der Rekursionen"

# 1 Grundlagen

## 1.1 Allgemein

Die HIS (Herstellerinitiative Software) besteht aus den Automobilherstellern Audi, BMW Group, DaimlerChrysler, Porsche und Volkswagen und hat sich in fünf Arbeitskreisen organisiert, deren Ziel das Erarbeiten einheitlicher Standards bei den Themen Standard-Softwaremodule für Netzwerke, Prozessreifegradermittlung, Softwaretest, Softwaretools und Programmieren von Steuergeräten ist.

Software-Metriken sind die Grundlage für effizientes Projekt- und Qualitätsmanagement. Mit Software-Metriken können Aussagen über die Qualität des Software-Produkts und des Software-Entwicklungsprozesses gemacht werden.

Die HIS legt in diesem Dokument eine Grundmenge von Metriken zur Bewertung von Software fest.

## 1.2 Referenzierte Dokumente

**Tabelle 1-1 Referenzierte Dokumente**

Dokument	Titel
HIS Subset MISRA C 1.0.2	Gemeinsames Subset der MISRA C Guidelines Stand: 22.03.2004; Version: 1.0.2
ISO/IEC 9899:1999	Programming languages - C Stand: 1999
Definition „Zyklomatische Komplexität“	Complexity Metric (Arthur H. Watson, Thomas J. McCabe; Computer Systems Laboratory; National Institute of Standards and Technology; Gaithersburg, MD 20899-0001; United States of America) Stand: August 1996 <a href="http://hissa.nist.gov/HHRFdata/Artifacts/ITLdoc/235/chapter2.htm">http://hissa.nist.gov/HHRFdata/Artifacts/ITLdoc/235/chapter2.htm</a>

## 1.3 Glossar

### Metrik

Ein Maß (Metrik) ist die objektive Zuordnung eines Wertes zu einer Entität, um ein spezifisches Merkmal zu charakterisieren.

(Quelle: Deutschsprachige Anwendergruppe für Software-Metrik und Aufwandschätzung e.V.  
<http://www.dasma.org>)

### Statement

Eine einzeln auszuführende Aktion.

(Quelle [ISO/IEC 9899:1999]; Kapitel 6.8: „A statement specifies an action to be performed.“)

## 2 Umsetzung

### 2.1 Erhebung der Metriken

Die festgelegten Metriken werden auf Basis „compilierbare Einheit“ erhoben<sup>1</sup>. Für jede Funktion ist ein eigener Metrikensatz zu erheben.

### 2.2 Anwendung der Metriken

Der Software-Lieferant ist verantwortlich für die Dokumentation der Messwerte gemäß der in diesem Dokument definierten Software Metriken.

Es gibt eine prinzipielle Unterscheidung in Metriken, die nur dokumentiert werden müssen (Tabelle 3-2 Metriken ohne Grenzwerte) und in Metriken (Tabelle 3-1 Metriken mit Grenzwerten), bei denen eine Verletzung der Grenzwerte zu weiteren Aktionen führt.

Alle Metriken sind zu berichten, alle Verletzungen der definierten Grenzen sind auf Funktionsniveau zu begründen.

Die Behebung von Verstößen gegen die Grenzen der Metriken ist anwendungsspezifisch (z.B. Funktionssoftware, Betriebssystem) und OEM-spezifisch. Jeder OEM kann zum Beispiel außerhalb der in Tabelle 3-1 definierten Grenzen („Grün-Bereich“) weitere Bereiche festlegen, in denen er die Software vorbehaltlich von Nachbesserungen annimmt („Gelb-Bereich“), oder komplett ablehnt („Rot-Bereich“).

---

<sup>1</sup> Für verschiedene Analysetools gilt „compilierbare Einheit“ = „Gesamtsoftware“, auch wenn nicht aufgelöste externe Schnittstellen auf das Gegenteil (Teilumfang) hinweisen, d.h. Grenzwerte können sich abhängig von „Gesamtsoftware“ oder „Teilumfang“ unterscheiden.

### 3 Metriken – Maßzahlen

#### 3.1 Metriken mit Grenzwerten

**Tabelle 3-1 Metriken mit Grenzwerten**

<i>Metrik</i>	<i>Beschreibung</i>	<i>Kommentar</i>	<i>Bereich</i>
Kommentardichte "COMF"	Verhältnis Anzahl der Kommentare (außerhalb und innerhalb von Funktionen) zur Anzahl der Statements	Verständlichkeit, Einarbeitung in den Code. Bei Verletzung entscheidet Dokumentation über Annahme / Freigabe. Nach Definition ist ein Wert > 1 nicht möglich.  Bezeichnung Logiscope: COMF = (BCOM + BCOB) / STMT  Bezeichnung QA-C: COMF = (STM28 / STM22)	> 0,2
Anzahl der Pfade "PATH"	Anzahl der nicht zyklischen Ausführungspfade (d.h. minimale Anzahl benötigter Testfälle)	Maßnahmen zur Verringerung von PATH: Aufteilung in mehrere Funktionen, Auslagern in Unterfunktionen.  Bezeichnung Logiscope: ct_path, PATH Bezeichnung QA-C: STPTH	1 - 80
Anzahl Sprunganweisungen "GOTO"	Anzahl der goto-Statements	Erhöht drastisch die Anzahl der Pfade (siehe PATH), reduziert die Testbarkeit  Bezeichnung Logiscope: ct_goto, GOTO Bezeichnung QA-C: STGTO	0
Zyklomatische Komplexität "v(G)"	Zyklomatische Nummer gemäß [Definition „Zyklomatische Komplexität“]	Maßnahmen zur Verringerung von v(G): Aufteilung in mehrere Funktionen, Auslagern in Unterfunktionen.  Bezeichnung Logiscope: ct_vg, VG, ct_cyclo Bezeichnung CodeSurfer: vG Bezeichnung QA-C: STCYC	1 - 10
Anzahl der aufrufenden Funktionen "CALLING"	Von wie vielen unterschiedlichen Funktionen wird diese Unterfunktion genutzt?	Bereich 1 bis 5 nur sinnvoll, wenn gesamtes System analysiert wird. Bei Teilsystemen/Bibliotheken ist 0 bis 5 zulässig. Bei Gesamtsystem gibt es die Ausnahme main(), da diese nur aus dem Startup-Code aufgerufen wird.  Bezeichnung Logiscope: dc_calling, NBCALLING Bezeichnung QA-C: STM29	0 - 5

<i>Metrik</i>	<i>Beschreibung</i>	<i>Kommentar</i>	<i>Bereich</i>
Anzahl der aufgerufenen Funktionen „CALLS“	Wie viele unterschiedliche Unterfunktionen ruft diese Funktion auf? Aufrufe derselben Unterfunktion zählen nur einfach	Überprüfung Verschachtelungsbreite  Bezeichnung Logiscope: dc_calls, DRCT_CALLS Bezeichnung QA-C: STCAL	0 - 7
Anzahl Funktionsparameter „PARAM“	Wie komplex ist das Interface einer Funktion?	Komplexität der Funktion, Stackbedarf. Strukturen und Arrays verbergen die Komplexität genauso.  Bezeichnung Logiscope: ic_param Bezeichnung QA-C: STPAR	0 - 5
Anzahl der Befehle pro Funktion „STMT“	Wie komplex ist die Funktion?	Leere Funktion fällt durch!  Bezeichnung Logiscope: lc_stat, STMT Bezeichnung QA-C: STST3	1 - 50
Anzahl der Aufrufebenen „LEVEL“	Verschachtelungstiefe einer Funktion.	Maximale Verschachtelungstiefe innerhalb einer Funktion +1  Bezeichnung Logiscope: LEVL Bezeichnung QA-C: STMIF	0 - 4
Anzahl der Aussprungpunkte „RETURN“	Wie viele Aussprungpunkte hat eine Funktion?	Komplexität der Funktion, Änderbarkeit der Funktion, Endekriterien. 0 = Funktion ohne explizites Return-Statement.  Bezeichnung Logiscope: RETU Bezeichnung QA-C: STM19	0 - 1
Stabilitätsindex „S <sub>i</sub> “	Der Stabilitätsindex liefert ein Maß für die Zahl der Veränderungen (Änderungen, Streichungen, Hinzufügungen) zwischen zwei Versionen einer Software.  $\text{Stabilitätsindex } S_i = (\text{STMT} - (S_{\text{change}} + S_{\text{new}} + S_{\text{del}})) / \text{STMT}$	Erhebung auf Funktionsebene oder auf Gesamtsoftware.  Der Wert ist definitionsgemäß kleiner oder gleich 1. Je mehr Veränderungen die neue Softwareversion enthält, desto niedriger wird dieser Wert. Wurden überhaupt keine Änderungen gemacht, ergibt diese Metrik den Wert 1.  Erwartungsgemäß ist der Stabilitätsindex für zwei frühe aufeinanderfolgende Versionen einer Software eher niedrig und steigt für die späteren Versionen immer weiter an.  Bezeichnung Logiscope: nicht vorhanden Bezeichnung QA-C: nicht vorhanden	≤ 1

<i>Metrik</i>	<i>Beschreibung</i>	<i>Kommentar</i>	<i>Bereich</i>
Sprachumfang „VOCF“	Der Sprachumfang ist ein Indikator für den Aufwand zum Warten/Ändern von Funktionen. $\text{VOCF} = (N1 + N2) / (n1 + n2),$ wobei n1 = Anzahl verschiedener Operatoren N1 = Summe aller Operatoren n2 = Anzahl verschiedener Operanden N2 = Summe aller Operanden	Hoher Wert = ähnliche oder duplizierte Codeanteile; Auslagern in Unterfunktionen eventuell sinnvoll.  Bezeichnung Logiscope: VOCF  Bezeichnung QA-C: Ermittelt aus STOPN (n2), STOPT (n1), STM21 (N1), STM22 (N2)	1 - 4
„NOMV“	Gesamtzahl der Verletzungen der Regeln des [HIS Subset MISRA C 1.0.2].  NOMV = Number of MISRA HIS Subset violations	Einhaltung von HIS-Regeln bei der Erstellung der Software  Bezeichnung Logiscope: nicht vorhanden  Bezeichnung QA-C: Kann über den Compliance Report ermittelt werden.	0
„NOMVPR“	Anzahl der Verletzungen der Regeln des [HIS Subset MISRA C 1.0.2], aufgeschlüsselt nach Regeln.  NOMVRP = Number of MISRA violations per rule	Einhaltung von HIS-Regeln bei der Erstellung der Software Beinhaltet auch Regel zu „Dead Code“  Bezeichnung Logiscope: nicht vorhanden  Bezeichnung QA-C: Kann über den Compliance Report ermittelt werden.	0
Anzahl der Rekursionen „ap_cg_cycle“	Call graph recursions	Rekursion über eine oder mehrere Funktionen.  Bezeichnung Logiscope: ap_cg_cycle , GA_CYCLE  Bezeichnung QA-C: STNRA	0

### 3.2 Metriken ohne Grenzwerte

Die in folgender Tabelle aufgeführten Metriken stellen reine Messgrößen dar und müssen auf jeden Fall dokumentiert werden.

**Tabelle 3-2 Metriken ohne Grenzwerte**

Bezeichnung	Beschreibung	Kommentar	Bereich
„S <sub>change</sub> “	STMT(geändert) Anzahl der Statements einer Software, die zwischen der vorhergehenden und der aktuellen Version geändert wurden.	Bewertung des Änderungsumfangs seit der letzten offiziellen Softwarelieferung.  Dieser Wert wird zur Berechnung des Stabilitätsindex S <sub>i</sub> benötigt.  Bezeichnung Logiscope: nicht vorhanden Bezeichnung QA-C: nicht vorhanden	—
„S <sub>del</sub> “	STMT(gelöscht) Anzahl der Statements einer Software, die zwischen der vorhergehenden und der aktuellen Version gelöscht wurden.	Bewertung des Änderungsumfangs seit der letzten offiziellen Softwarelieferung  Dieser Wert wird zur Berechnung des Stabilitätsindex S <sub>i</sub> benötigt.  Bezeichnung Logiscope: nicht vorhanden Bezeichnung QA-C: nicht vorhanden	—
„S <sub>new</sub> “	STMT(neu) Anzahl der Statements einer Software, die zwischen zwei Software-Versionen hinzugefügt wurden.	Bewertung des Änderungsumfangs seit der letzten offiziellen Softwarelieferung  Dieser Wert wird zur Berechnung des Stabilitätsindex S <sub>i</sub> benötigt.  Bezeichnung Logiscope: nicht vorhanden Bezeichnung QA-C: nicht vorhanden	—