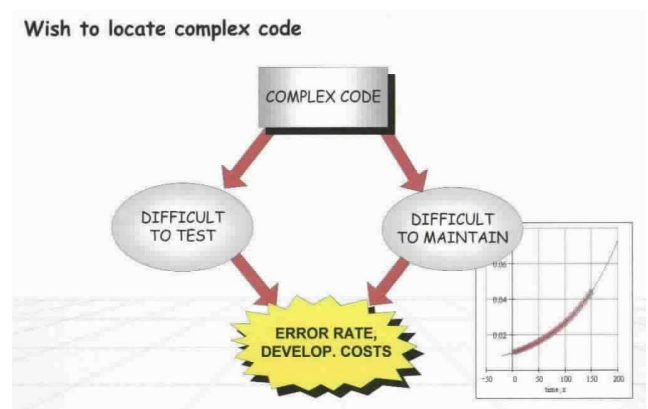


## Testwell CMT++/CMTJava Softwarekomplexitätsmessungen für C/C++ und Java-Anwendungen

### Warum Softwarekomplexitätsmessungen?

- die Softwarekomplexität (Code Complexity) korreliert mit der Fehlerrate und der "Robustheit" einer Anwendung
- unnötig komplexer Code ist oft Ursache schlechter Softwarequalität und fehlerhafter Programme.
- komplexer Code ist schwierig zu testen -> die Anwendung hat mehr Bugs
- komplexer Code ist schwierig zu warten
- der Einsatz der Komplexitätsmesstools CMT++ bzw. CMTJava hilft Ihnen Kosten einzusparen



### Was wird mit CMT++ bzw. CMTJava gemessen?

Basierend auf einer statischen Analyse Ihrer Anwendungen gibt CMT++ bzw. CMTJava eine Einschätzung dafür wie fehlerhaft Ihr Programm wahrscheinlich ist, wieviel Zeit notwendig ist um den Code zu verstehen, was das logische Volumen der Software ist usw.

Das Tool deckt Codeteile auf, die intensiv getestet werden sollen.

### CMT++ und CMTJava zeigen die folgenden Softwaremetriken an:

#### a) Zeilenmetriken/Lines-of-code (LOC) Metriken

LOCbl	Anzahl leerer Programmzeilen
LOCcom	Anzahl der Programmierzeilen mit Kommentaren
LOCphy	Gesamtanzahl der Zeilen (number of physical lines)
LOCpro	Anzahl der Zeilen mit Programmcode

#### b) McCabe Cyclomatic Number $v(G)$

Die Cyclomatic Number  $v(G)$  beschreibt die Komplexität des Steuerflusses einer Anwendung.

### c) Halstead-Metriken

B	geschätzte Fehleranzahl (estimated number of errors)
D	Schwierigkeitsgrad, Fehlerträchtigkeit (difficulty level, error proneness)
E	Implementierungsaufwand (effort to implement)
L	Programmiveau, Abstraktionsniveau des Programms (program level, abstraction level of the program)
N	Programmlänge
N1	Gesamtanzahl der Operatoren
N2	Gesamtanzahl der Operanden
n	Vokabulargröße (unterschiedliche Operatoren + unterschiedliche Operanden)
n1	Anzahl unterschiedlicher Operatoren
n2	Anzahl unterschiedlicher Operanden
T	Implementierungszeit / Zeit um einen Programmteil zu verstehen
V	Programmvolumen bzw. Informationsgehalt des Programms bzw. eines Algorithmus

### d) Wartungsaufwand (Maintainability Index)

Der Wartungsindex zeigt Ihnen unter anderem an, wann es kostengünstiger bzw. einfacher ist Programmteile neu zu schreiben anstatt bestehenden zu komplexen Code zu modifizieren.

Die von CMT++ bzw. CMTJava vorgegebenen Alarmlimits basieren auf langjährigen Erfahrungen mit Softwareprojekten, deren Qualität und Wartbarkeit als vorbildlich gilt.

Testwell Softwarekomplexitätsmeßtools verwenden weltweit anerkannte Metriken (Industriestandards), die in Forschungsprojekten in den USA erstellt worden sind (Software Engineering Institute of Carnegie Mellon University).

Die "akzeptierbare Codekomplexität" kann allerdings je nach Projekt unterschiedlich sein. CMT++ und CMTJava erlauben dem Nutzer daher eine Anpassung der Alarmdefinitionen an die jeweiligen Projektanforderungen im Control-Panel des Tools.

### *CMT++/CMTJava Highlights :*

- analysiert ursprüngliche nicht präprozessierte Dateien und Projekte
- extrem schnelle Analyse: komplette, selbst umfangreiche Projekten werden in wenigen Minuten analysiert
- arbeitet problemlos mit vielen und großen Dateien
- Alarmlimits können an Projektanforderungen angepaßt werden
- Ausgaben in HTML-, XML- oder Textformat (erlauben die Weiterverarbeitung der Daten, z.Bsp. mit Excel)
- GUI-Integration in Visual C++ Developer Studio
- Verfügbar für verschiedenen Plattformen: Windows, Linux, HP, Solaris



**Verifysoft Technology GmbH**  
Technologiepark, In der Spöck 10  
D-77656 Offenburg  
Tel. +49 781 6392 027  
<http://www.verifysoft.com>

