
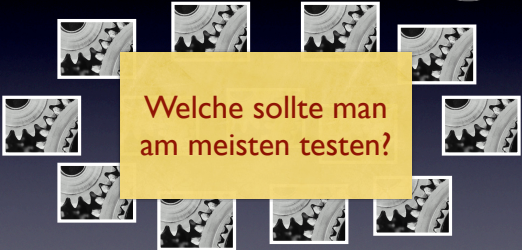


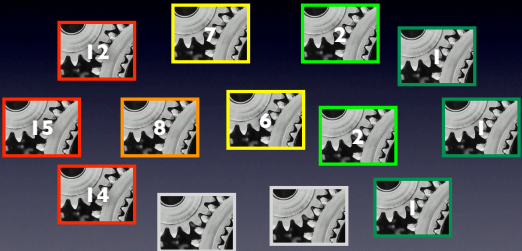
Module



Welche sollte man am meisten testen?



Früheres Fehlschlagen



Früheres Fehlschlagen

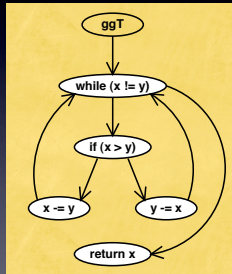


Komplexitäts-Metriken

- Messen die «Komplexität» des Codes
- Beispiel:
 - #Lines – Anzahl der Code-Zeilen
 - #Classes – Anzahl der Klassen
 - #Parameters – Anzahl der Parameter
- Je mehr, je komplexer!

McCabe-Metrik

- Misst Komplexität des Kontrollflusses
- $V(G) = e - n + 2p$
- e: #Übergänge
- n: #Anweisungen
- p: #Eingänge



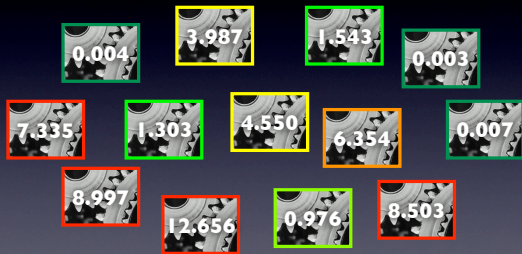
Wartbarkeits-Index

Größe Vokabular McCabe-Komplexität

$$\text{Wartbarkeit} = 171 - 5.2 \ln(\overline{V}) - 0.23 \sqrt{\overline{V(G)}} - 16.2 \ln(\overline{L}) + 50 \sin\left(\sqrt{2.4 \overline{C}}\right)$$

Anteil Kommentarzeilen Codezeilen

Komplexitäts-Metriken



Womit können wir Fehlschläge vorhersagen?



Beides kombinieren

Frühere Fehler



- Wir sammeln Fehlverhalten beim Anwender innerhalb der ersten 6 Monate nach Freigabe
- Wie bilden die Fehler ab auf Codestellen und somit auf *Module (Binaries)*
- Wir wissen so, wie *fehlerträchtig* ein Modul ist

Beides kombinieren

Frühere Fehler



Metriken

#Lines	#Vars	#Classes
#Params	#Reads	#Writes
#Arcs	#Blocks	McCabe
Fan In	Fan Out	...

Beides kombinieren

Frühere Fehler



Metriken

#Lines	#Vars	#Classes	#Lines	#Vars	#Classes	#Lines	#Vars	#Classes
#Params	#Reads	#Writes	#Params	#Reads	#Writes	#Params	#Reads	#Writes
#Arcs	#Blocks	McCabe	#Arcs	#Blocks	McCabe	#Arcs	#Blocks	McCabe
Fan In	Fan Out	...	Fan In	Fan Out	...	Fan In	Fan Out	...

Beides kombinieren

Frühere Fehler



Metriken

#Lines	#Vars	#Classes	#Lines	#Vars	#Classes	#Lines	#Vars	#Classes
#Params	#Reads	#Writes	#Params	#Reads	#Writes	#Params	#Reads	#Writes
#Arcs	#Blocks	McCabe	#Arcs	#Blocks	McCabe	#Arcs	#Blocks	McCabe
Fan In	Fan Out	...	Fan In	Fan Out	...	Fan In	Fan Out	...

Korrelation




McCabe ~ Fehler

Fehler vorhersagen



Korrelation McCabe ~ Fehler

Projekte

- Internet Explorer 6 
- IIS Server 
- Windows Process Messaging 
- DirectX
- NetMeeting 

> 1.000.000 Codezeilen

Projekte

A B C D E

Fragestellungen

- Korrelieren Metriken mit Fehlverhalten?
- Gibt es *universelle* Metriken, die für alle Projekte anwendbar sind?
- Können wir *Fehlerorte* (= fehlerträchtige Module) vorhersagen?

Korrelieren Metriken mit Fehlverhalten?

Projekt	Korrelierte Metriken
A	#Classes und 5 abgeleitete
B	fast alle
C	alle außer <i>MaxInheritanceDepth</i>
D	nur #Lines
E	#Functions, #Arcs, McCabe

Korrelieren Metriken mit Fehlverhalten?

Projekt	Korrelierte Metriken
A	#Classes und 5 abgeleitete
B	fast alle
C	alle außer <i>MaxInheritanceDepth</i>
D	nur #Lines
E	#Functions, #Arcs, McCabe

JA

Gibt es universelle Metriken?

Projekt	Korrelierte Metriken
A	#Classes und 5 abgeleitete
B	fast alle
C	alle außer <i>MaxInheritanceDepth</i>
D	nur #Lines
E	#Functions, #Arcs, McCabe

Gibt es universelle Metriken?

Projekt	Korrelierte Metriken
A	#Classes und 5 abgeleitete
B	fast alle
C	alle außer <i>MaxInheritanceDepth</i>
D	nur #Lines
E	#Functions, #Arcs, McCabe

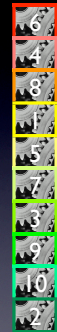
NEIN

Vorhersage fehlerträchtiger Module

- Grundidee: *Metriken kombinieren*
- Größtes Gewicht für *vorhersagende Metriken*
- Problem: Metriken sind *untereinander korreliert*
- Lösung: *Principal Component Analysis (PCA)*

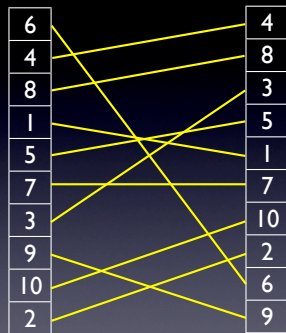
Vorhersagekraft

- Aus den Hauptkomponenten können wir *Regressions-Modelle* ableiten
- In diese können wir Metriken stecken, um die Fehlerwahrscheinlichkeit vorherzusagen
- Module können anhand der Fehlerwahrscheinlichkeit *sortiert* werden



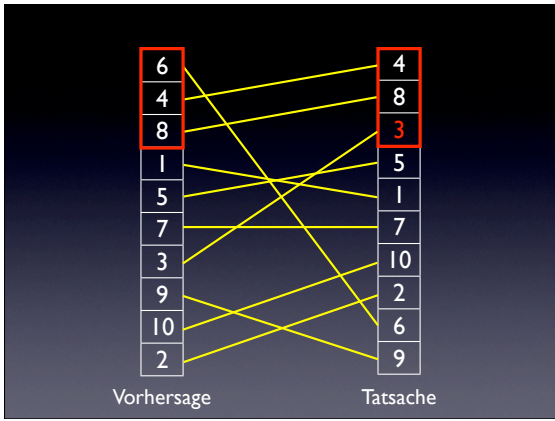
Eine Sortierung

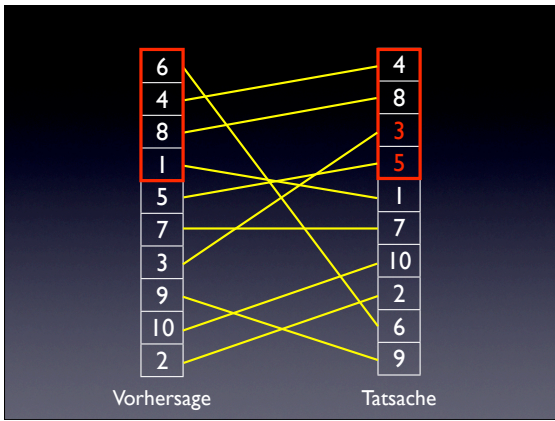
- 1/3 der Module
- sortiert anhand eines Prädiktors, der aus 2/3 der Module stammt
- kann mit tatsächlicher Sortierung verglichen werden

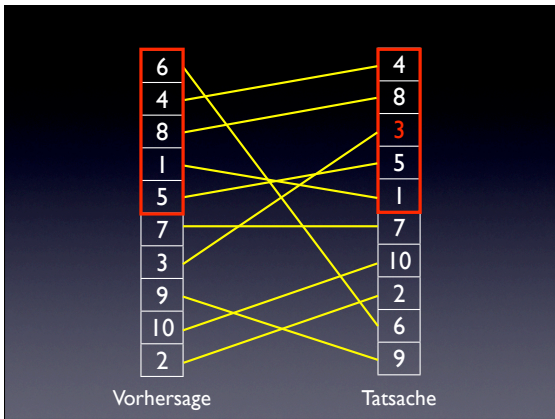


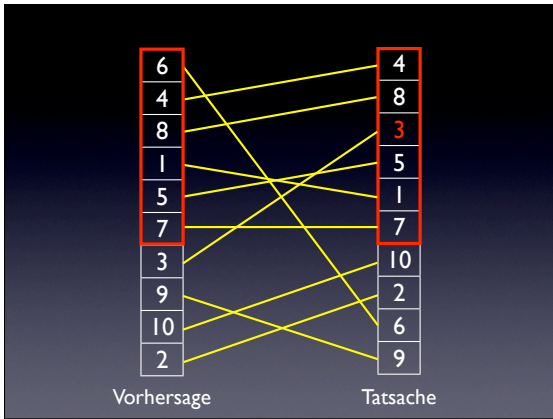
Vorhersage

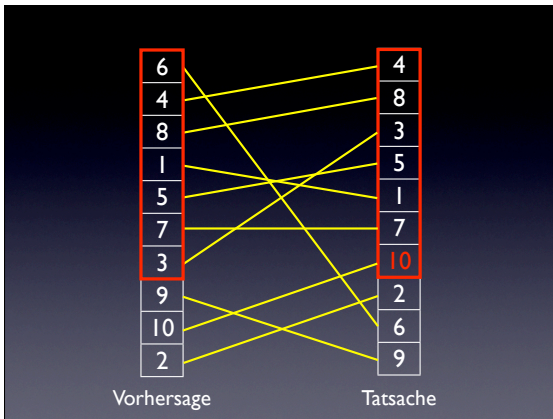
Tatsache











Können wir Fehlerorte vorhersagen?

Projekt	#Komponenten	R ² -Wert
A	9	0.741
B	6	0.779
C	7	0.579
D	7	0.684
E	5	0.919

Können wir Fehlerorte vorhersagen?

Projekt	#Komponenten	R ² -Wert
A	9	0.741
B	6	0.779
C	7	0.579
D	7	0.684
E	5	0.919

JA

Sind Prädiktoren übertragbar?

Project	A	B	C	D	E
A	•	no	no	no	no
B	no	•	yes	no	no
C	no	yes	•	no	(yes)
D	no	no	no	•	no
E	no	no	(yes)	no	•

Sind Prädiktoren übertragbar?

Project	A	B	C	D	E
A	•	no	no	no	no
B	no	•	yes	no	no
C	no	yes	•	no	(yes)
D	no	no	no	•	no
E	no	no	(yes)	no	•

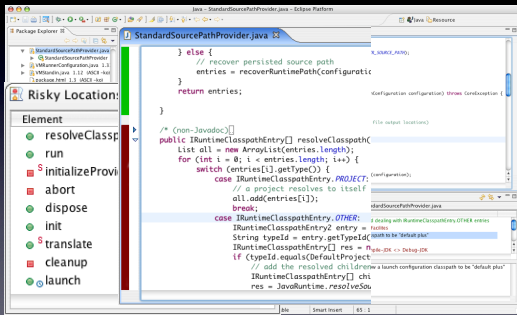
JEIN

Ausblick

Mehr Metriken	Mehr Projekte
Mehr Werkzeuge	Mehr Daten

– Einsatz für Windows Vista –

Hatari



Zusammenfassung

- ★ Metriken korrelieren mit Fehlerverhalten
- ★ Es gibt keine universellen Metriken, die für alle Projekte anwendbar wären
- ★ Wir können fehlerträchtige Module vorhersagen
- ★ Software-Archive erzählen viel über reale Fehler
- ★ Goldgrube für Data Mining und Evaluierung