Installationsanleitung
und erste Schritte
für das Eclipse-Plugin „Metrics“

1. Einführung:

Metrics ist ein Plugin für Eclipse, dass die Erfassung und Analyse der quantitativen Eigenschaften eines Projektes (LOC, Vererbungstiefe, Anzahl Methoden, etc.) erleichtert. Ihr könnt es im Rahmen des Praktikums benutzen um die Metriken zu errechnen die ihr im Auswertungsbogen angeben müsst.

2. Installation:

Es gibt zwei Möglichkeiten PlugIns in Eclipse zu installieren. Entweder ihr macht es manuell, oder ihr lasst Eclipse sich automatisch updaten.

2.1. Installation über Eclipse:

-startet Eclipse, und geht über:

Help-Menü -> Software Updates -> Find and Install...

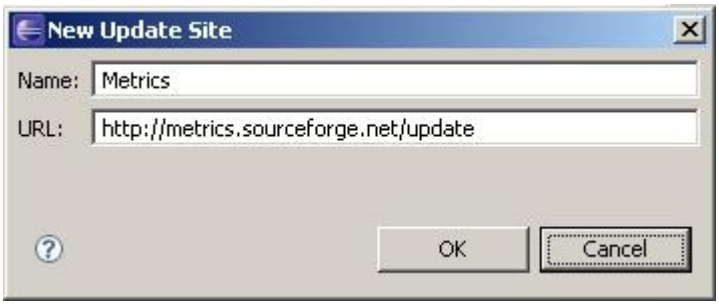
-wählt im nächsten Fenster:

Search for new features to install.

-dann wählt ihr:

new Remote Site

und gebt als url http://metrics.sourceforge.net/update **an**.



-klickt auf

Finish und folgt den Installationsanweisungen.

2.2. Manuell:

-ladet euch das Plugin herunter, ihr findet es auf
http://metrics.sourceforge.net/update

-entpackt das ganze.

-kopiert die jar-Datei aus dem **features**-Ordner und in den features-Ordner von Eclipse

-kopiert die jar-Datei im **plugins**-Ordner in den plugins-Ordner von Eclipse

-damit das Plugin erkannt wird müsst ihr **Eclipse neu starten**

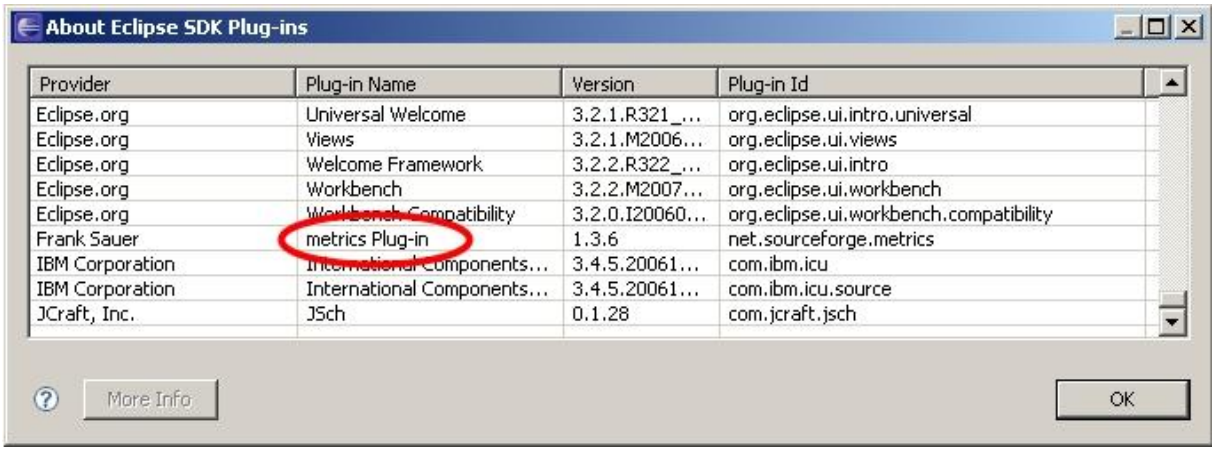
Wichtig: Das Plugin setzt **mindestens Eclipse 3.1.** voraus!

So könnt ihr prüfen ob das Plugin korrekt installiert wurde:

-geht über
Help -> About Eclipse SDK

-und klickt dann auf
Plug-in Details.

Das Plugin müsste jetzt in der Liste vorhanden sein.



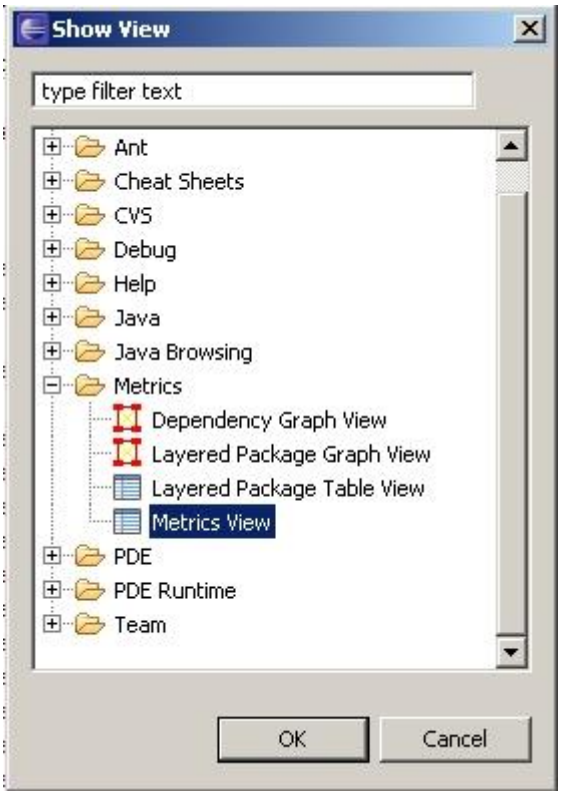
3. Verwendung:

-geht über:

Window -> Show View -> Other...

und wählt dann:

Metrics/Metrics View aus.



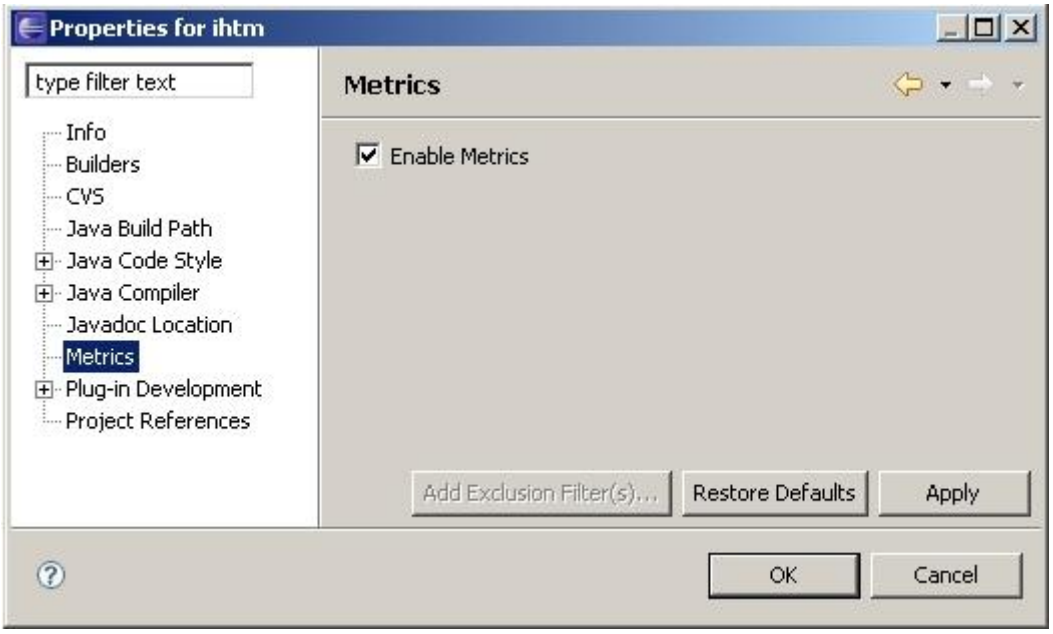
Jetzt wird eine neue View angelegt, es wird aber noch nichts berechnet.
Das müsst ihr erst für euer Projekt aktivieren:

-geht dazu in den **Package-Explorer** und:

rechtsklickt auf euer Projekt -> Properties.

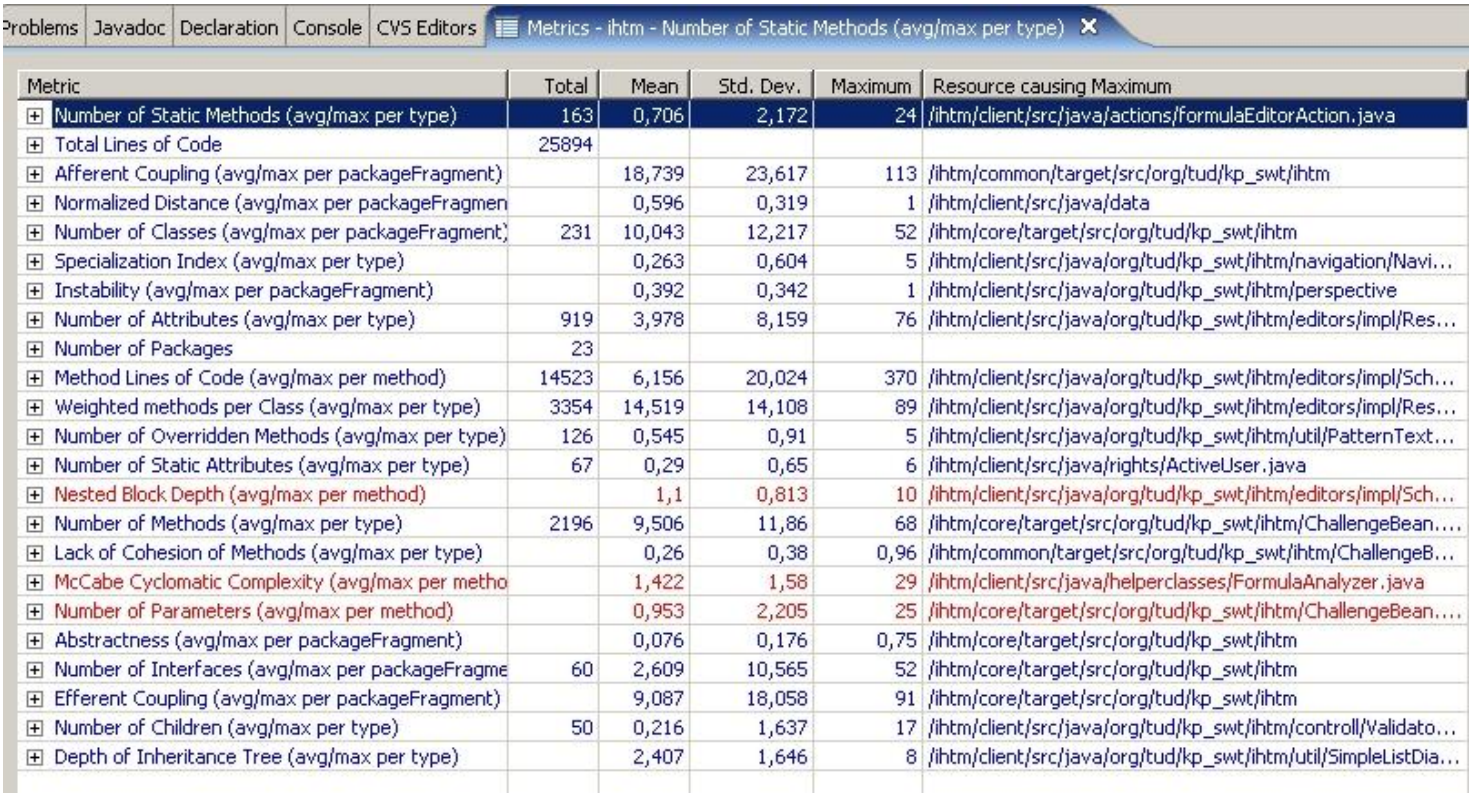
-im Properties-Dialog aktiviert ihr im Menüpunkt **Metrics**:

die **Enable Metrics**-Checkbox



Wenn ihr jetzt einen **rebuild** ausführt, werden die Metriken für euer Projekt berechnet und in der View angezeigt. Die Werte werden von nun an bei jedem rebuild aktualisiert.

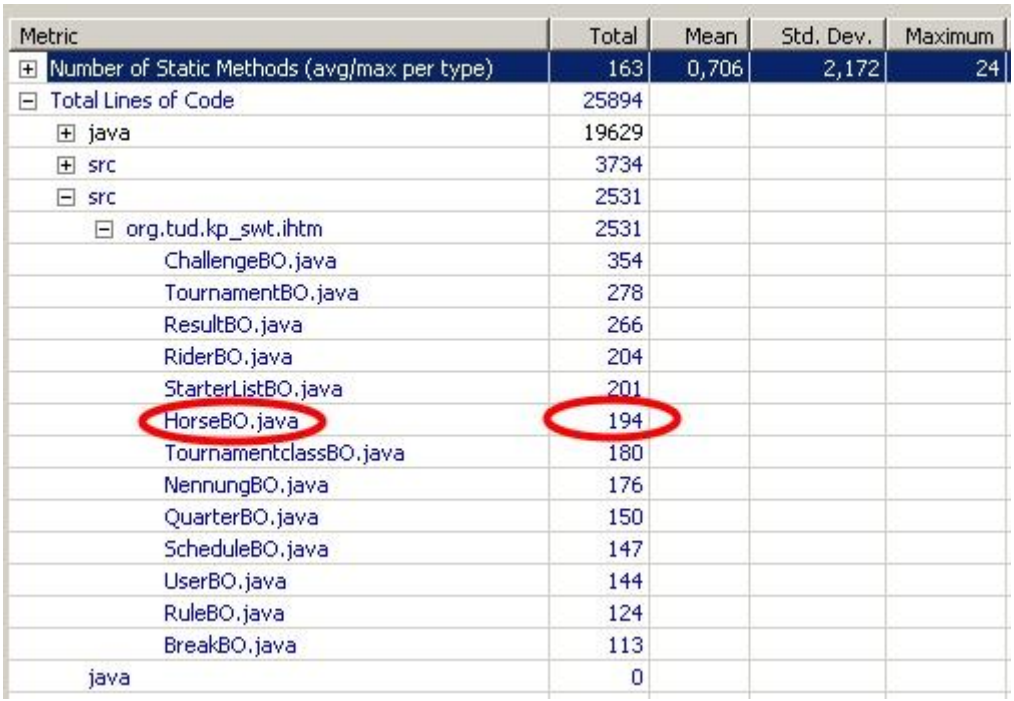
Nach der Berechnung sieht die Metrics-View ungefähr so aus:



Problems | Javadoc | Declaration | Console | CVS Editors | Metrics - ihtm - Number of Static Methods (avg/max per type)

Metric	Total	Mean	Std. Dev.	Maximum	Resource causing Maximum
Number of Static Methods (avg/max per type)	163	0,706	2,172	24	/ihtm/client/src/java/actions/formulaEditorAction.java
Total Lines of Code	25894				
Afferent Coupling (avg/max per packageFragment)		18,739	23,617	113	/ihtm/common/target/src/org/tud/kp_swt/ihtm
Normalized Distance (avg/max per packageFragmen		0,596	0,319	1	/ihtm/client/src/java/data
Number of Classes (avg/max per packageFragment)	231	10,043	12,217	52	/ihtm/core/target/src/org/tud/kp_swt/ihtm
Specialization Index (avg/max per type)		0,263	0,604	5	/ihtm/client/src/java/org/tud/kp_swt/ihtm/navigation/Navi...
Instability (avg/max per packageFragment)		0,392	0,342	1	/ihtm/client/src/java/org/tud/kp_swt/ihtm/perspective
Number of Attributes (avg/max per type)	919	3,978	8,159	76	/ihtm/client/src/java/org/tud/kp_swt/ihtm/editors/impl/Res...
Number of Packages	23				
Method Lines of Code (avg/max per method)	14523	6,156	20,024	370	/ihtm/client/src/java/org/tud/kp_swt/ihtm/editors/impl/Sch...
Weighted methods per Class (avg/max per type)	3354	14,519	14,108	89	/ihtm/client/src/java/org/tud/kp_swt/ihtm/editors/impl/Res...
Number of Overridden Methods (avg/max per type)	126	0,545	0,91	5	/ihtm/client/src/java/org/tud/kp_swt/ihtm/util/PatternText...
Number of Static Attributes (avg/max per type)	67	0,29	0,65	6	/ihtm/client/src/java/rights/ActiveUser.java
Nested Block Depth (avg/max per method)		1,1	0,813	10	/ihtm/client/src/java/org/tud/kp_swt/ihtm/editors/impl/Sch...
Number of Methods (avg/max per type)	2196	9,506	11,86	68	/ihtm/core/target/src/org/tud/kp_swt/ihtm/ChallengeBean....
Lack of Cohesion of Methods (avg/max per type)		0,26	0,38	0,96	/ihtm/common/target/src/org/tud/kp_swt/ihtm/ChallengeB...
McCabe Cyclomatic Complexity (avg/max per metho		1,422	1,58	29	/ihtm/client/src/java/helperclasses/FormulaAnalyzer.java
Number of Parameters (avg/max per method)		0,953	2,205	25	/ihtm/core/target/src/org/tud/kp_swt/ihtm/ChallengeBean....
Abstractness (avg/max per packageFragment)		0,076	0,176	0,75	/ihtm/core/target/src/org/tud/kp_swt/ihtm
Number of Interfaces (avg/max per packageFragme	60	2,609	10,565	52	/ihtm/core/target/src/org/tud/kp_swt/ihtm
Efferent Coupling (avg/max per packageFragment)		9,087	18,058	91	/ihtm/core/target/src/org/tud/kp_swt/ihtm
Number of Children (avg/max per type)	50	0,216	1,637	17	/ihtm/client/src/java/org/tud/kp_swt/ihtm/controll/Validato...
Depth of Inheritance Tree (avg/max per type)		2,407	1,646	8	/ihtm/client/src/java/org/tud/kp_swt/ihtm/util/SimpleListDia...

Auf Metriken, die blau aufgelistet sind kann man doppelklicken um zu der Ressource (Klasse, Methode,...) zu navigieren, die den Maximalwert verursacht. Die einzelnen Metriken sind als Baumstrukturen aufgelistet, die man immer weiter aufsplitten kann um zu sehen mit welcher Gewichtung bestimmte Ressourcen in den Gesamtwert einfließen.



Metric	Total	Mean	Std. Dev.	Maximum
⊞ Number of Static Methods (avg/max per type)	163	0,706	2,172	24
⊟ Total Lines of Code	25894			
⊞ java	19629			
⊞ src	3734			
⊟ src	2531			
⊟ org.tud.kp_swt.ihtm	2531			
ChallengeBO.java	354			
TournamentBO.java	278			
ResultBO.java	266			
RiderBO.java	204			
StarterListBO.java	201			
HorseBO.java	194			
TournamentclassBO.java	180			
NennungBO.java	176			
QuarterBO.java	150			
ScheduleBO.java	147			
UserBO.java	144			
RuleBO.java	124			
BreakBO.java	113			
java	0			

Bsp.: Von den 25894 Codezeilen, die das Projekt groß ist, enthält die Klasse HorseBO: 194.

4. relevante Metriken:

Das sind die Metriken, die ihr braucht um den Fragebogen auszufüllen:

4.1. Gesamtzahl Codezeilen ohne Leer- und Kommentarzeilen:

-die Zeile: Total Lines of Code (Total-Spalte)

4.2. Anzahl der Klassen:

-die Zeile: Number of Classes (Total-Spalte)

4.3. Methoden pro Klasse:

Max: Number of Methods (Maximum-Spalte)

Durchschnitt: Number of Methods (Mean-Spalte)

4.4. Kodezeilen pro Methode:

Max: Method Lines of Code (Maximum-Spalte)

Durchschnitt: Method Lines of Code (Mean-Spalte)

4.5. Vererbung:

maximale Tiefe: Depth of Inheritance Tree (Maximum-Spalte)

maximale Breite: Number of Children (Maximum-Spalte)