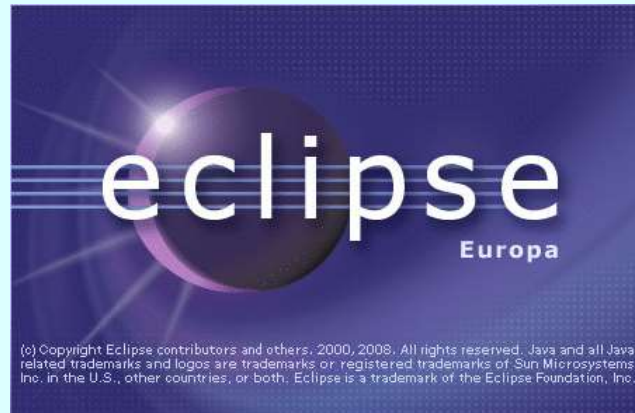


Python und Geoprocessing mit Eclipse und PyDev



Yvo Weidmann

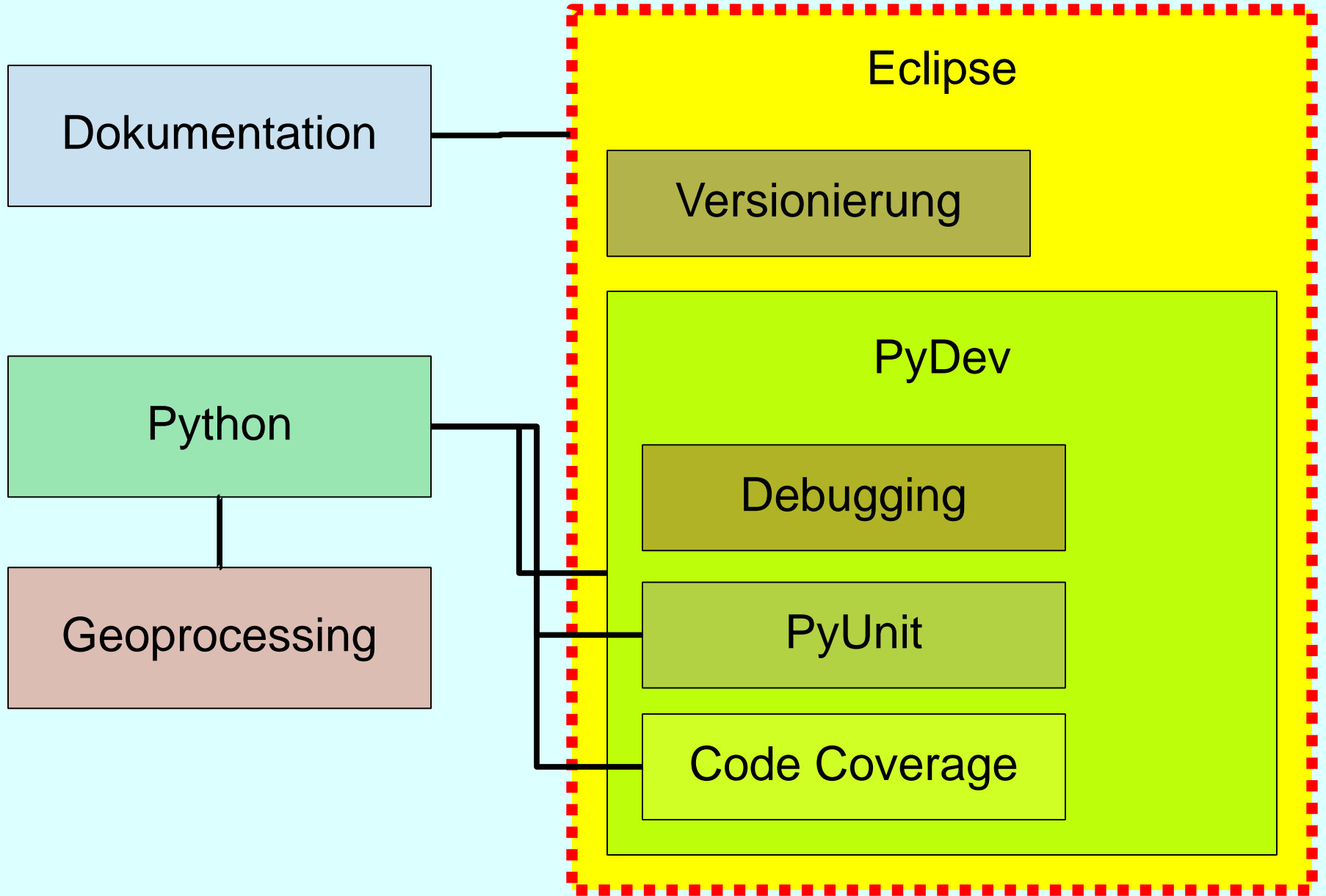
Dipl. Ing. Geomatik FHNW

MAS Software Engineering FHB

Inhalt

- Eclipse, Bezug und Installation
- PyDev
 - Geoprocessing
 - PyUnit
 - Debugging
 - Code Coverage
- Versionierung
- Dokumentation
- Zusammenfassung

Übersichtskarte



Eclipse

Eclipse ist eine integrierte Software
Entwicklungsumgebung (IDE)

Eclipse Project: 2001 von IBM und verschiedensten
Software Herstellern gegründet

Eclipse Foundation: 2004 als unabhängige not-for-
profit reorganisiert

Breit abgestützte Open-Source Community in
Entwicklung und Nutzung

Quasi-Industriestandard in Java-Entwicklung

Framework für verschiedenste Programmiersprachen

Unzählige Open-Source und kommerzielle Plugins

Direkt von Eclipse

Download verschiedensten Konfigurationen von
<http://www.eclipse.org>

Freier Eclipse Download Service

Download individuellen Konfigurationen von
<http://ondemand.yoxos.com/geteclipse/start>

Eclipse > Bezug > yoxos

Yoxos OnDemand - Get your personalized Eclipse - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://ondemand.yoxos.com/geteclipse/start

Library Version: latest

Windows Linux Mac OS X

free eclipse download service

Plug-In Explorer Profiles Schedule

Search: Enter at least 3 characters

Advanced search

- Programming Languages
 - CAL Eclipse Plug-in ★★★★★
 - Cohatoc Examples ★★★★★
 - Cohatoc Runtime ★★★★★
 - Cohatoc SDK ★★★★★
 - Cute Launcher ★★★★★
 - Dynamic Languages - Ruby ★★★★★
 - Dynamic Languages Toolkit ★★★★★
 - Dynamic Languages Toolkit ★★★★★
 - Dynamic Languages Toolkit ★★★★★
 - Dynamic Languages Toolkit ★★★★★
 - EPIC ★★★★★
 - Eclipse Colorer ★★★★★
 - Eclipse Eiffel Development 1 ★★★★★
 - Eclipse Mono Integration ★★★★★
 - Erlang IDE ★★★★★
 - FP: Haskell support for Edclp ★★★★★
 - Grails Eclipse Feature ★★★★★
 - GroovyFeature ★★★★★
 - JAutodoc ★★★★★
 - JavaFX Development Tools ★★★★★
 - LunarEclipse ★★★★★
 - OcaIDE Feature ★★★★★
 - Perlipse ★★★★★
 - PyDev Extensions ★★★★★
 - PyDev for Eclipse ★★★★★
 - Scala Development Tools (1) ★★★★★

the following features will be added to your download for win32

- Java Development
 - Eclipse Java Development Test
 - Programming Languages
 - PyDev for Eclipse
 - Language binaries
- Runtime
 - Eclipse Help System
 - Eclipse Platform
 - Eclipse RCP
 - Equinox p2 Provisioning
- Source Code Management
 - Subclipse
 - MS Visual Source Safe (VSS) in Eclipse

Estimated download size: 83.79 MB

Save As Profile Start Download

Information License

General

| | | | | | |
|---------|-------------------------------|-------------|----------|---------------|---------|
| Name | Subclipse | Popularity: | ★★★★★ | | |
| ID | org.tigris.subversion.subclip | Provider | tigris.c | Download Size | 3.83 MB |
| Version | 1.4.4 | Environment | | | |

send my feedback >>>

Auspacken der von yoxos bereitgestellten und heruntergeladenen ZIP-Datei

Benötigt keinen Installationsvorgang auf dem lokalen Rechner

Aktuelle Java-Laufzeitumgebung ist Bedingung

Verschiedene Konfigurationen parallel auf dem lokalen Rechner lauffähig

PlugIns über *Find and Install* direkt aus Eclipse installierbar

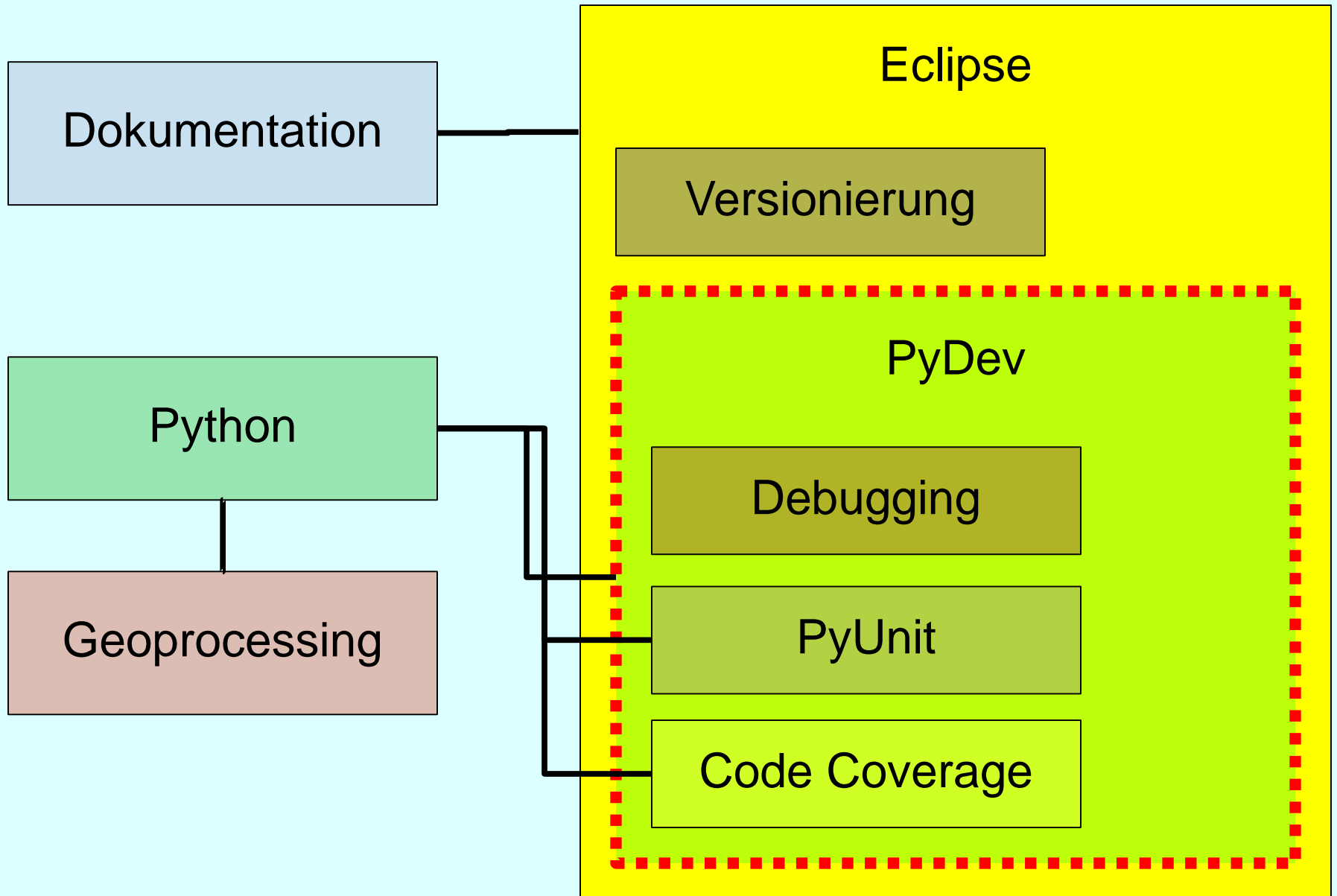
Eclipse > IDE

The screenshot displays the Eclipse IDE interface with the following components:

- Project Explorer (left):** Shows the project structure for 'Pydev WGWGeoprocessor'. It includes folders like 'src', 'data', and 'documentation', and files such as 'TableHandling.py', 'TableHandlingTest.py', and 'TableExceptions.py'.
- Code Editor (center):** Displays the source code for 'TableHandling.py'. The code defines a class 'CTableField' with various attributes and methods, including a constructor and a private getter.
- Object Catalog (right):** Shows a hierarchical view of the classes and methods defined in the project, including 'Geoprocessor', 'Parameter', 'TableExceptions', and 'CTableField'.
- Console (bottom):** Displays a list of tasks or messages, including 'TC00: Ab Version 2.5 die Kontrollstruktur mit except-Block verfeinern' and 'TC00: Typ des Feldes noch einbeziehen'.

Red dashed boxes highlight the Project Explorer, Code Editor, and Object Catalog. The text 'Objekt Katalog' is written in red over the Object Catalog, and 'Editor' is written in red over the Code Editor. The text 'Projekt Explorer' is written in red over the Project Explorer. The text 'Weiter Info (Tasks, Console, ...)' is written in red over the Console.

Übersichtskarte



Open-Source PlugIn in Eclipse

Entwicklung von Python-Programmen

<http://pydev.sourceforge.net/>

Einige Features:

- Interaktive Python Konsole (ab Version 1.3.19)

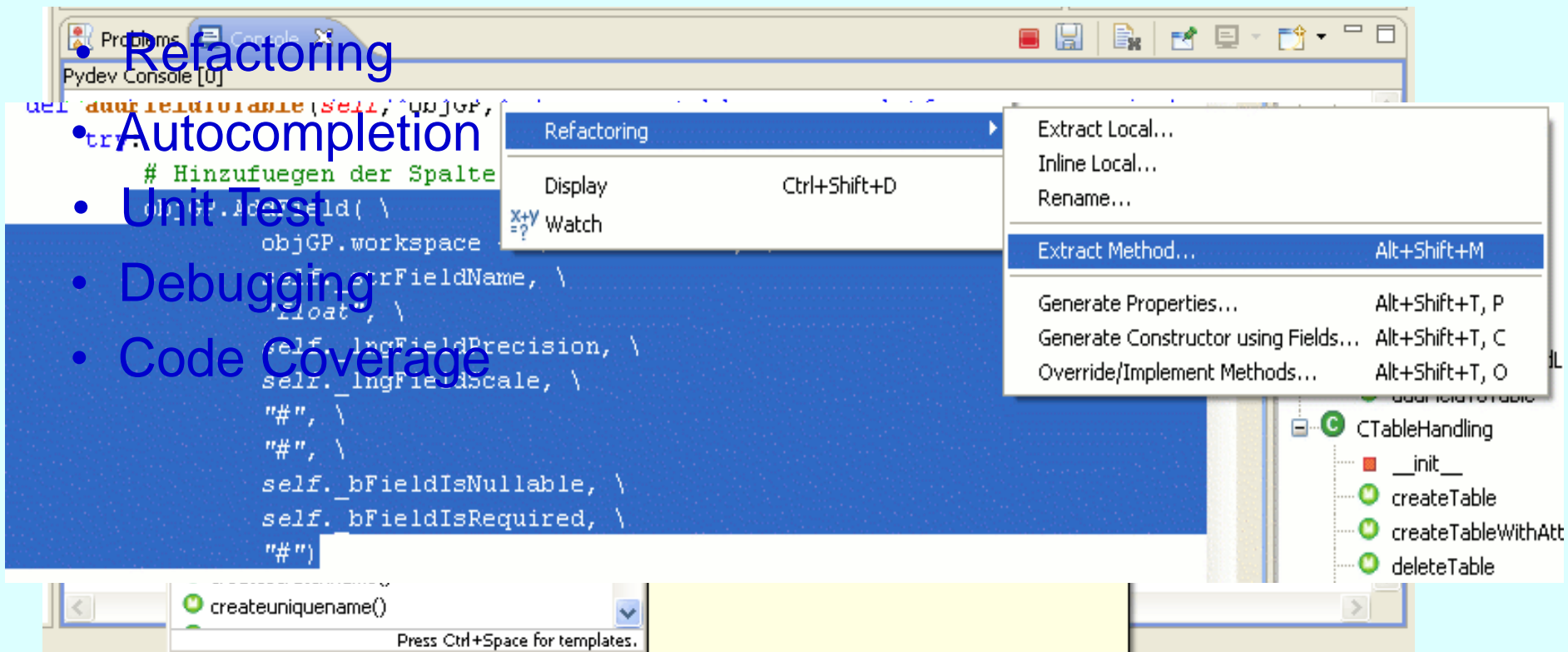
- Refactoring

- Autocompletion

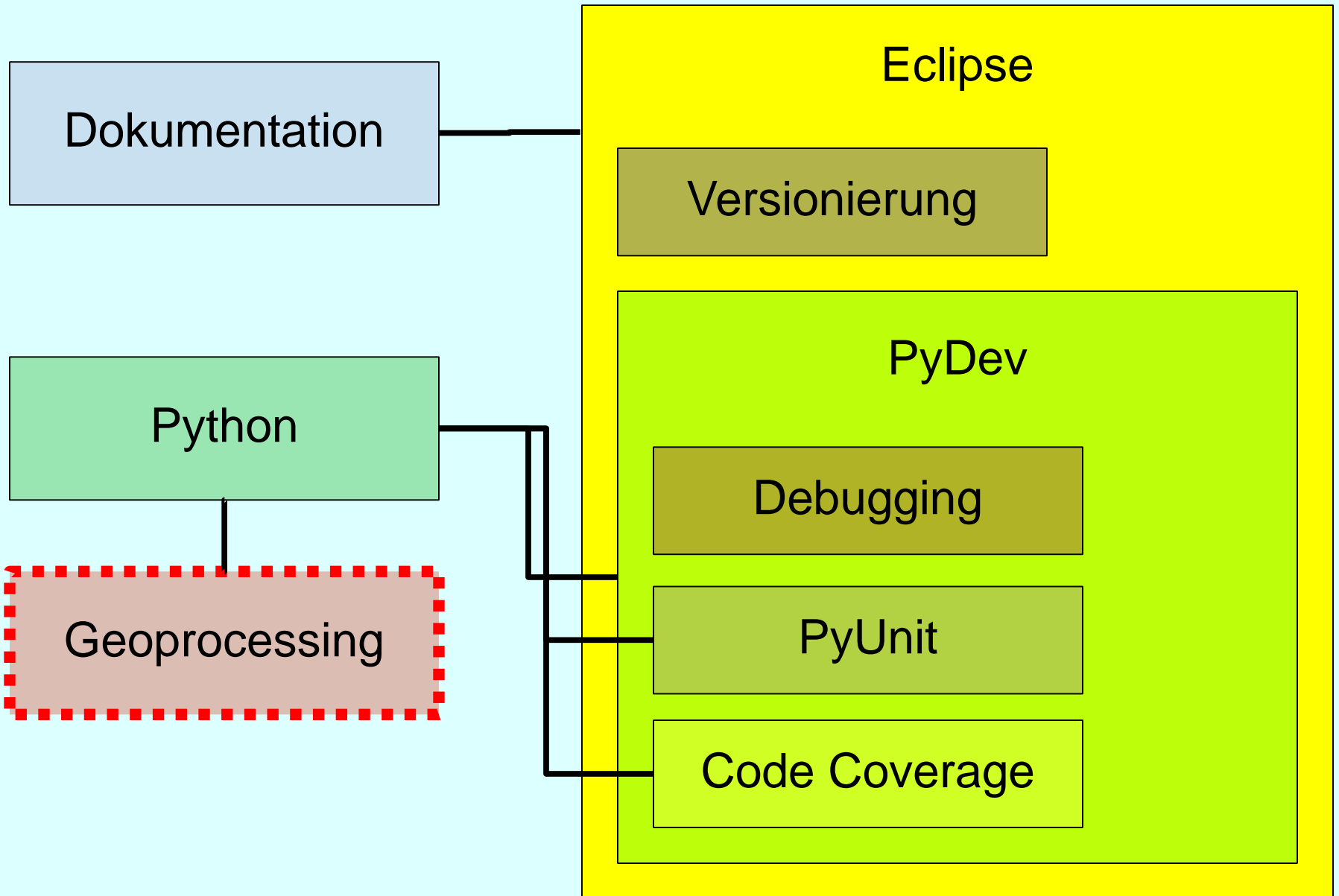
- Unit Test

- Debugging

- Code Coverage



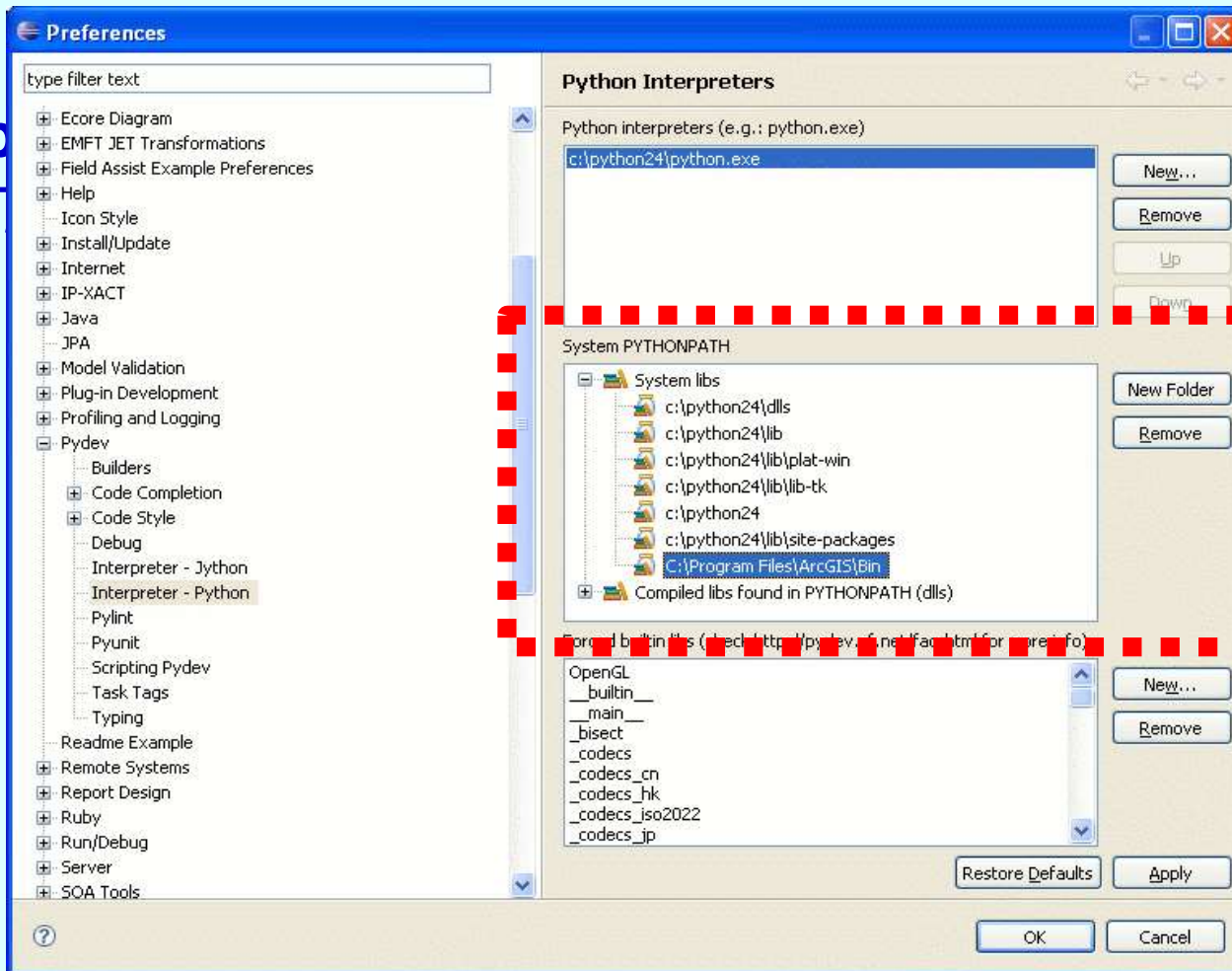
Übersichtskarte



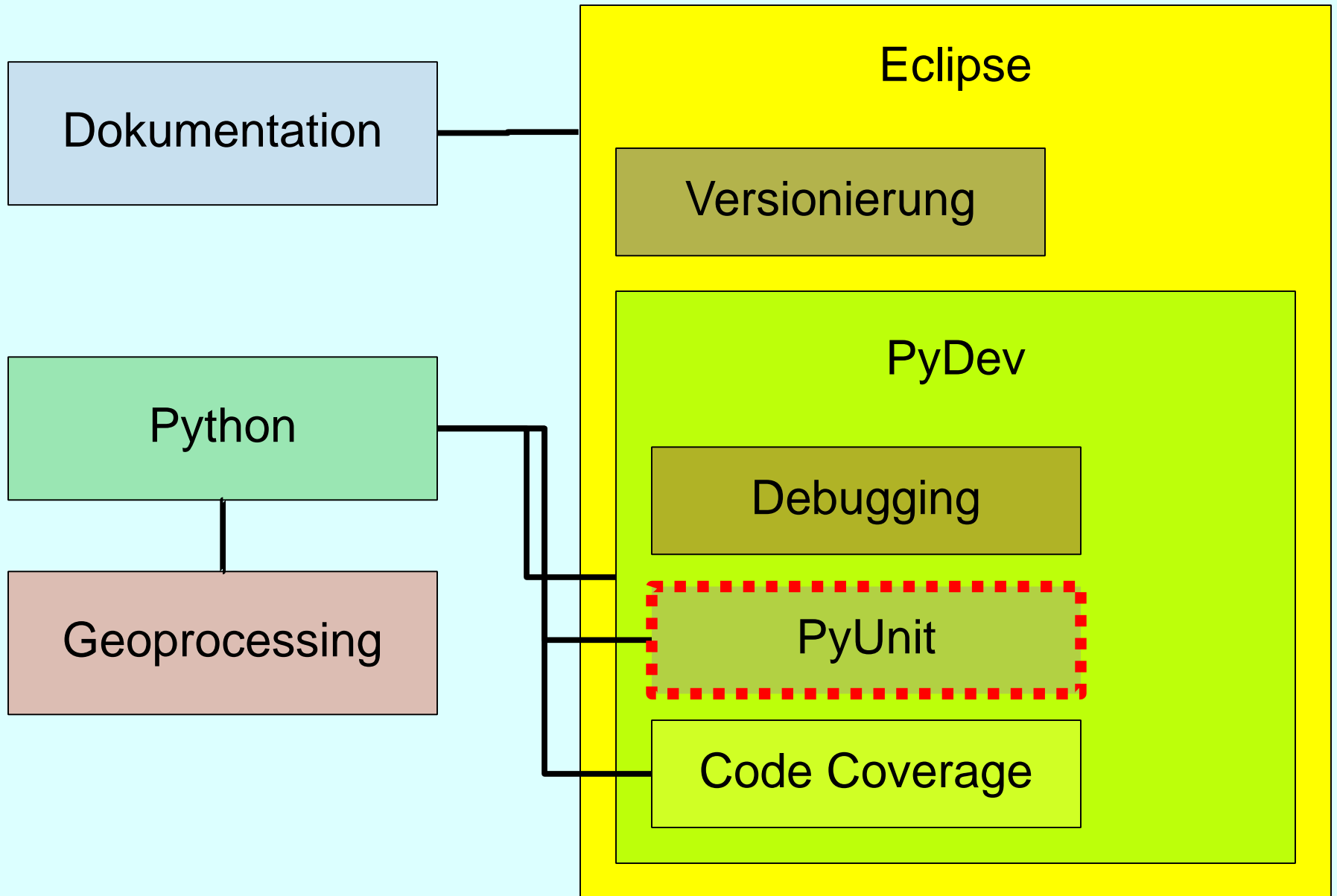
ArcGIS Geoprocessor mit PyDev

Das *Geoprocessor Programming Model* kann direkt verwendet werden.

Vorbereitung
PYT



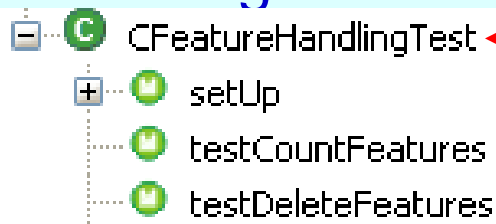
Übersichtskarte



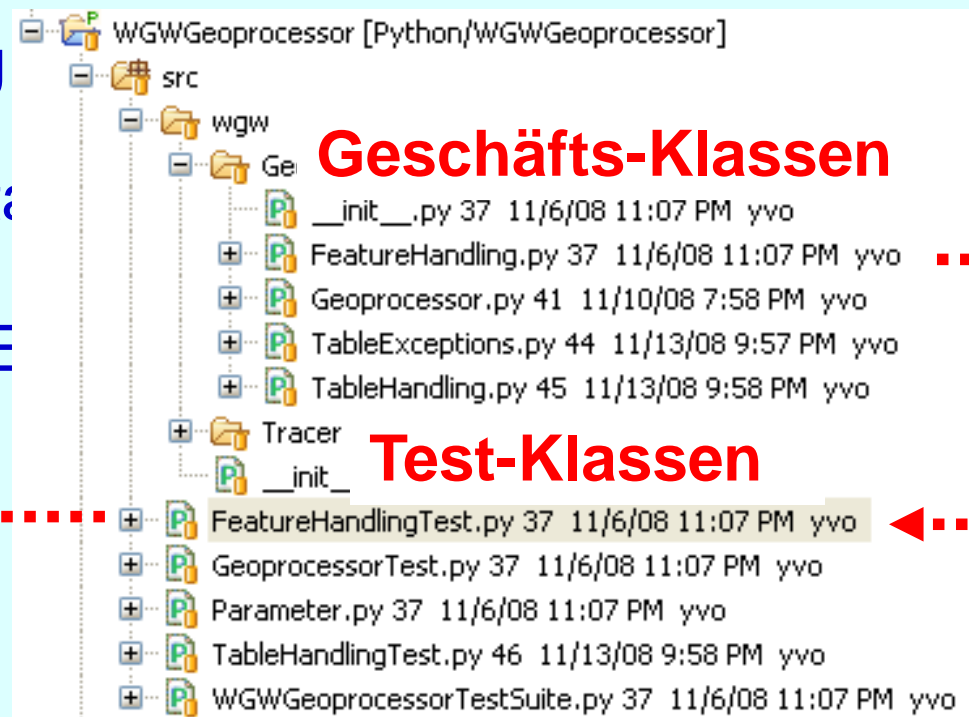
Testgetriebene Entwicklung

PyDev integriert die Unit-Test Umgebung *PyUnit*. Diese kann für normale Python- und Geoprocessing-Anwendungen verwendet werden.

- Implementation von *Test-Klassen* und *-Methoden*
- Zusammenfassung
- Automatisierte Tests
- Ausgabe der Test-E



Test-Methoden



Durchführung

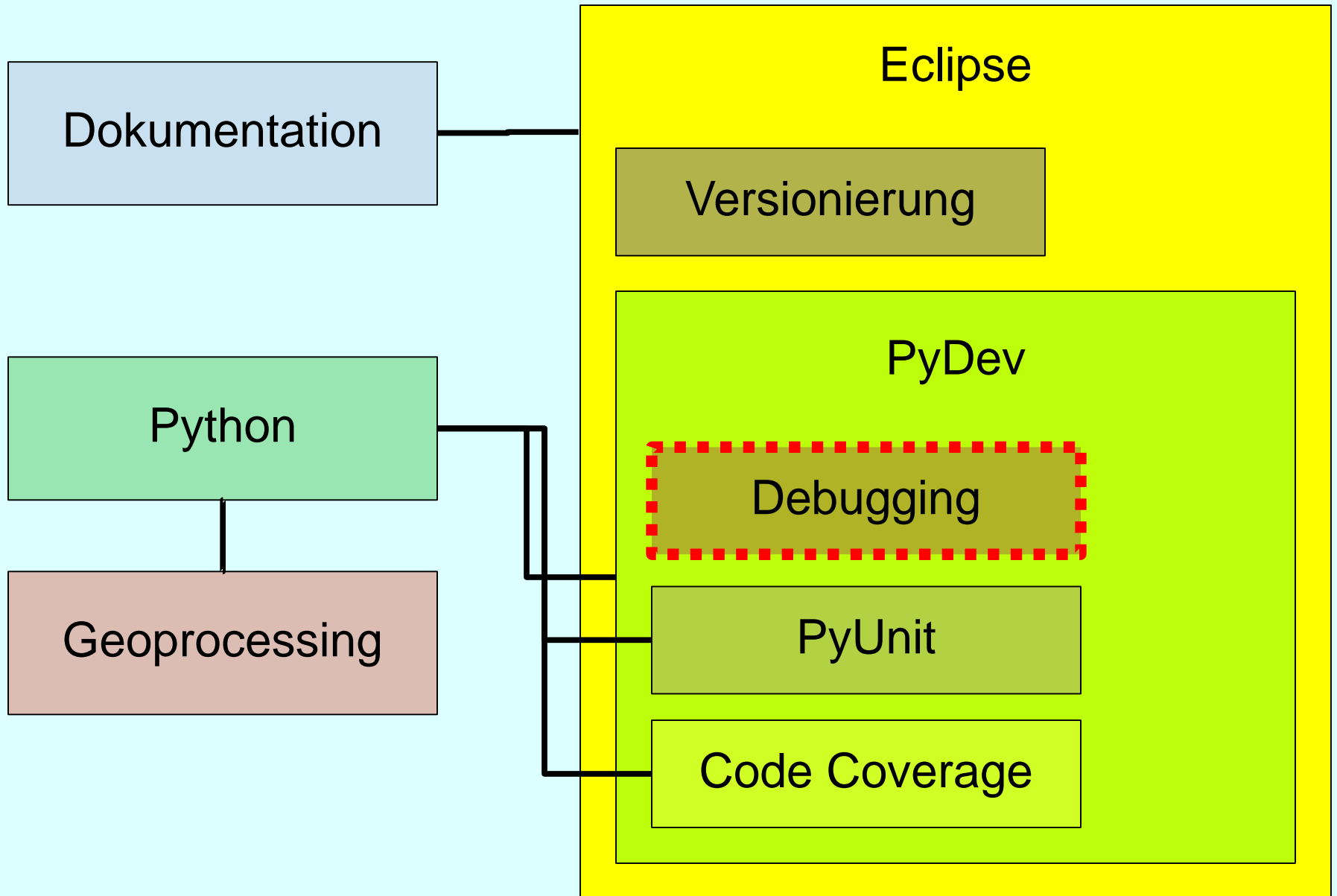
Nach der Implementation der *Test-Klassen* und *-Methoden* können die Tests automatisiert durchlaufen werden.

Ausgabe: Die Resultate werden in der Konsole ausgegeben.

```
testAreFieldsEquals (TableHandlingTest.CTableHandlingTest) ... ok
t testAreFieldsEquals (TableHandlingTest.CTableHandlingTest) ... ok
t testCTableParameter (TableHandlingTest.CTableHandlingTest) ... ok
t testCreateTable (TableHandlingTest.CTableHandlingTest) ... ok
t testCreateTableWA (TableHandlingTest.CTableHandlingTest) ... ok
  testToStringField (TableHandlingTest.CTableHandlingTest) ... ok
=
E -----
( ---
- Ran 5 tests in 114.813s
S
P OK
-
Ran 5 tests in 47.422s

FAILED (failures=1)
```

Übersichtskarte



Debugging von Python und Geoprocessing Projekten

PyDev integriert die Unit-Test Umgebung *PyUnit*. Diese kann für normale Python- und Geoprocessing-Anwendungen verwendet werden.

Einige Features:

- Breakpoints / Conditional Breakpoints
- Step In, Step Over, Step Return
- Anzeige der aktuellen Werte von Variablen

Vorbereitung: Jedes lauffähige Python-Modul kann mit dem Debugger durchlaufen werden.

PyDev > Debugging

Tools

Threads

Breakpoints

Variables

Editor mit Breakpoints und Debugger

```
14  ## <list type="bullet">
15  ## <item>Lesen aller Parameter fuer die Tests aus Parameter-Klasse</item>
16  ## <item>Instanzieren eines Objektes der zu testenden Klasse CTableHandling</item>
17  ## <item>Instanzieren eines Hilfsobjektes der Klasse CFeatureHandling</item>
18  ## </list>
19  ## </summary>
20  def setUp(self):
21
22      # Initialisieren der Parameter
23      self._objParameters = Parameter.CParameter()
24      self._strPgdb = self._objParameters.getPgdb()
25      self._strProduct = self._objParameters.getProduct()
26      self._objFH = egv.Geoprocessor.FeatureHandling.CFeatureHandling(self._strPgdb, self._strProduct)
27
28      # Instanzieren eines Objektes der zu testenden Klassen
29      self._objTableHandling = egv.Geoprocessor.TableHandling.CTableHandling(self._strPgdb, self._strProduct)
30
31      # Instanzieren eines Helfer-Objektes
32      self._objFH = egv.Geoprocessor.FeatureHandling.CFeatureHandling(self._strPgdb, self._strProduct)
33
34  ## <summary>
35  ## Runtersuchen der Testumgebung nach jedem Test.
36  ## </summary>
```

Global variables
Global variables
CTableHandlingTest: testCreateTable (TableHandlingTest.CT...

- Parameter
- unittest
- egv.Geoprocessor.FeatureHandling
- egv.Geoprocessor.TableExceptions
- egv.Geoprocessor.TableHandling
- CTableHandlingTest
 - setUp
 - self._objParameters
 - self._strPgdb
 - self._strTable
 - self._strProduct
 - self._objFH
 - self._objFH
 - tearDown
 - testCreateTable
 - testCreateTableWithAttributes
 - testToStringField
 - testAreFieldsEquals
 - testCTableFieldInvalidParameterException
 - testCTableFieldInvalidParameterMessage

pydev debugger
Finding files... [D:\02_WaterGisWeb)\Python\Geoprocessing\WGGeoprocessor\src\TableHandlingTest.py] ... done
Importing test modules ... done.

testAreFieldsEquals (TableHandlingTest.CTableHandlingTest) ... ok
testCTableFieldInvalidParameterException (TableHandlingTest.CTableHandlingTest) ... ok
testCTableFieldInvalidParameterMessage (TableHandlingTest.CTableHandlingTest) ... ok
testCreateTable (TableHandlingTest.CTableHandlingTest) ...

Variables

The screenshot shows the PyDev Variables window with a tree view of variables. The window is titled "(x)= Variables" and has a "Breakpoints" tab. The tree view is organized as follows:

- Globals
 - lstAttributes
 - objTextAttr1** (highlighted in yellow)
 - IsNullabe
 - IsRequired **Name**
 - Name
 - _CTableTextField__lngFieldLength
 - _bFieldIsNullabe
 - _bFieldIsRequired
 - _strFieldName** (highlighted in blue)
 - type
 - self
 - strTableAttribute

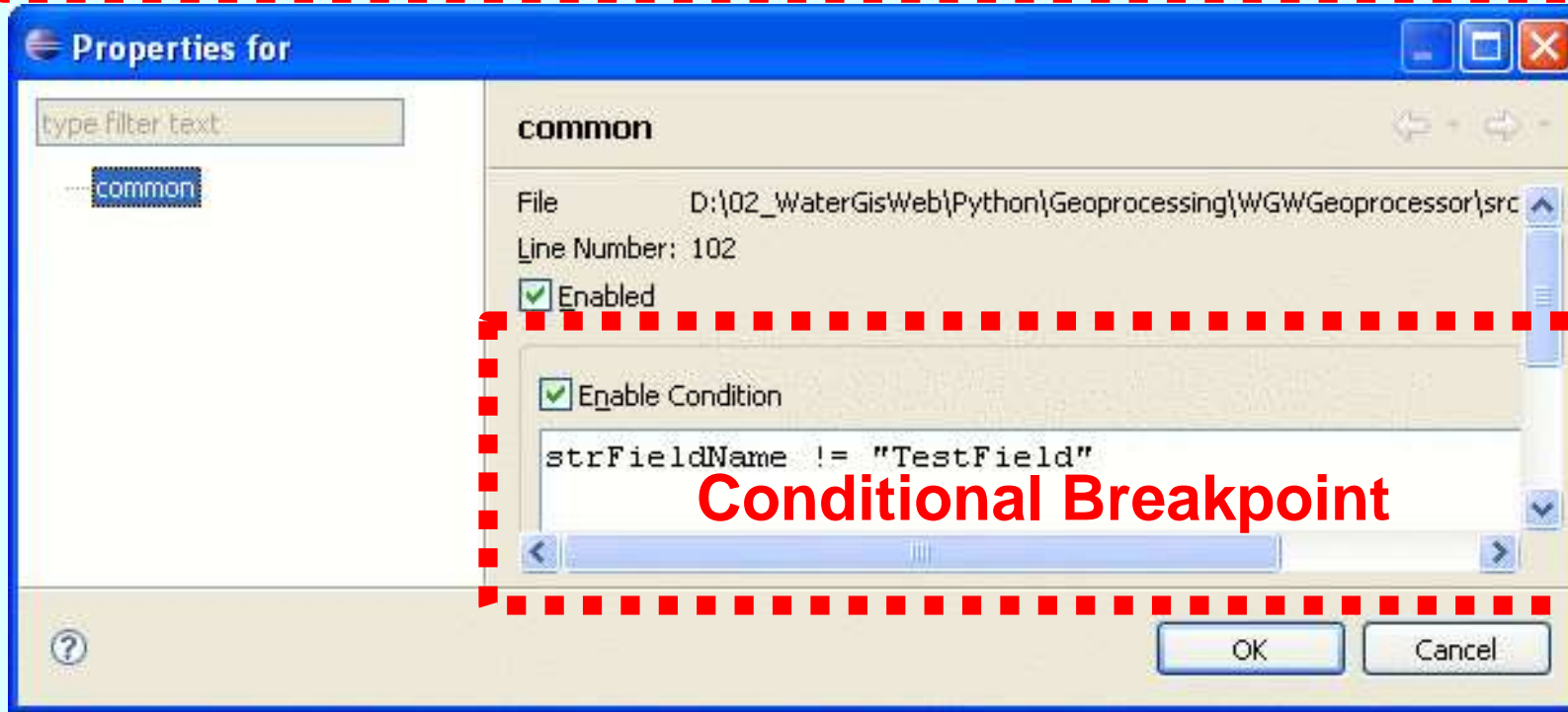
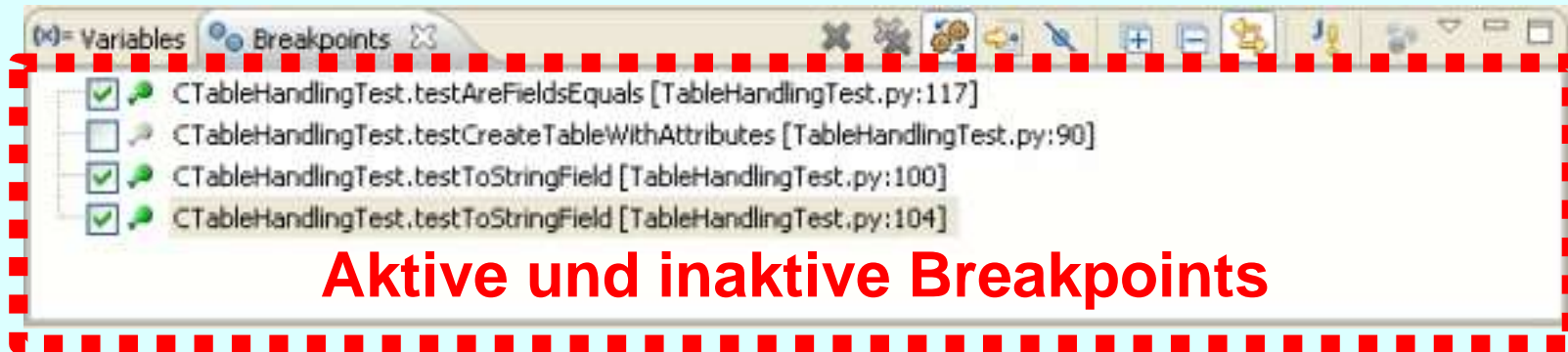
The right pane shows the values for the selected variables:

- Global variables
- lst: []
- CTableTextField: Text1** (highlighted in yellow)
- bool: True
- bool: True
- str: Text1
- int: 30
- bool: True
- bool: True
- str: Text1** (highlighted in blue)
- str: CTableTextField
- CTableHandlingTest: testCreateTableWithAttributes (TableH...
- str: TestTabelle_Attributes

At the bottom of the window, the text "str: Text1" is displayed.

Typ: Wert

Breakpoints



Schrittweises Debugging

Mit den Tools *Step In*, *Step Over* und *Step Return* lässt sich der Code schrittweise durchlaufen.

Toolbar



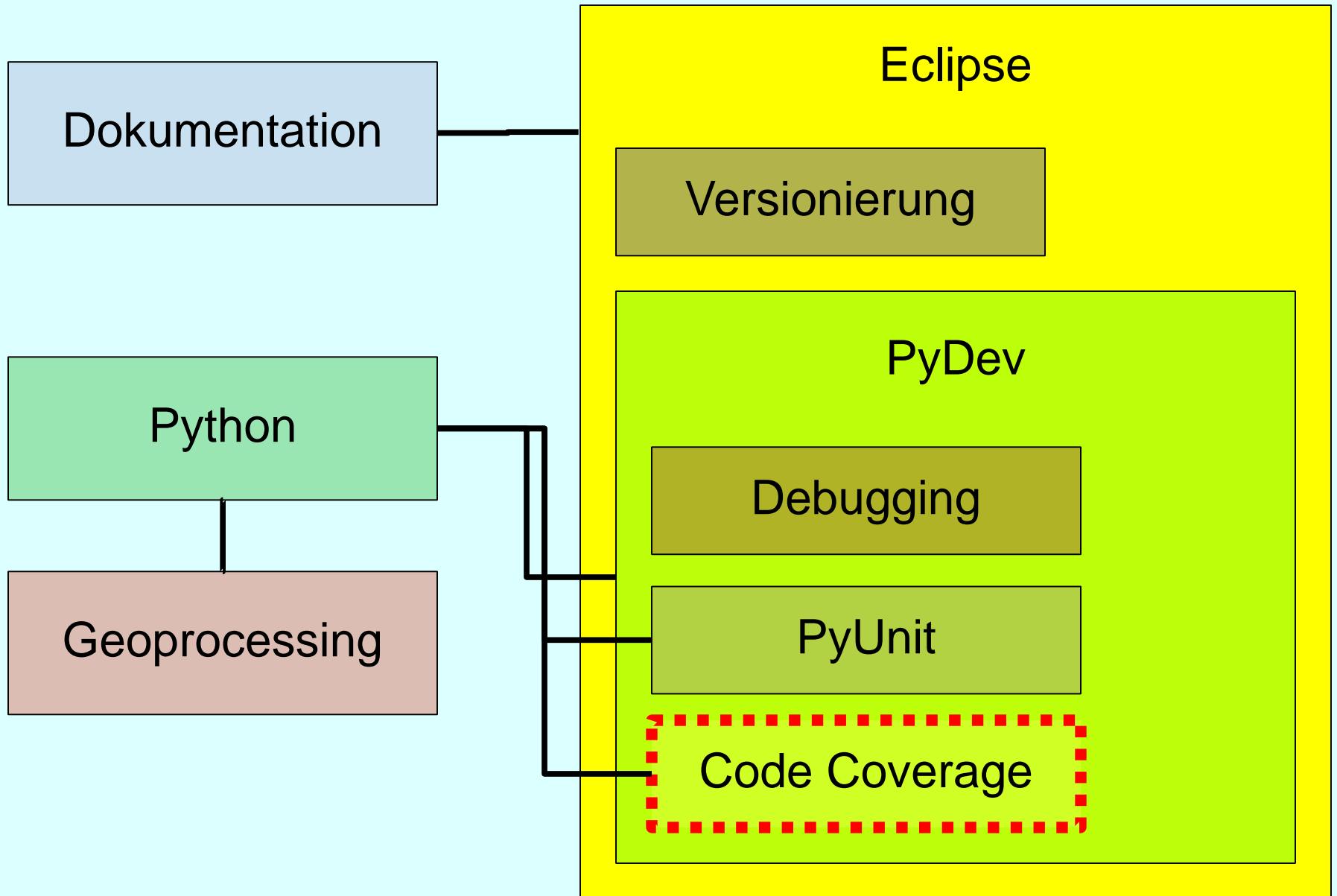
Debugging-Tools

Editor

```
22         # Initialisieren der Parameter
23         self.__objParameters = Parameter.CParameter()
24         self.__strPgdb = self.__objParameters.getPgdb()
25         self.__strTable = self.__objParameters.getTable()
26         self.__strProduct = self.__objParameters.getProduct()
```

Aktuelle Position des Debuggers

Übersichtskarte



Code Coverage

Mit der *Code Coverage* Analyse lassen sich beliebige Module analysieren

-> Anzeige von nicht verwendeten Code Zeilen und Anzahl der Ausführungen.

Anwendung:

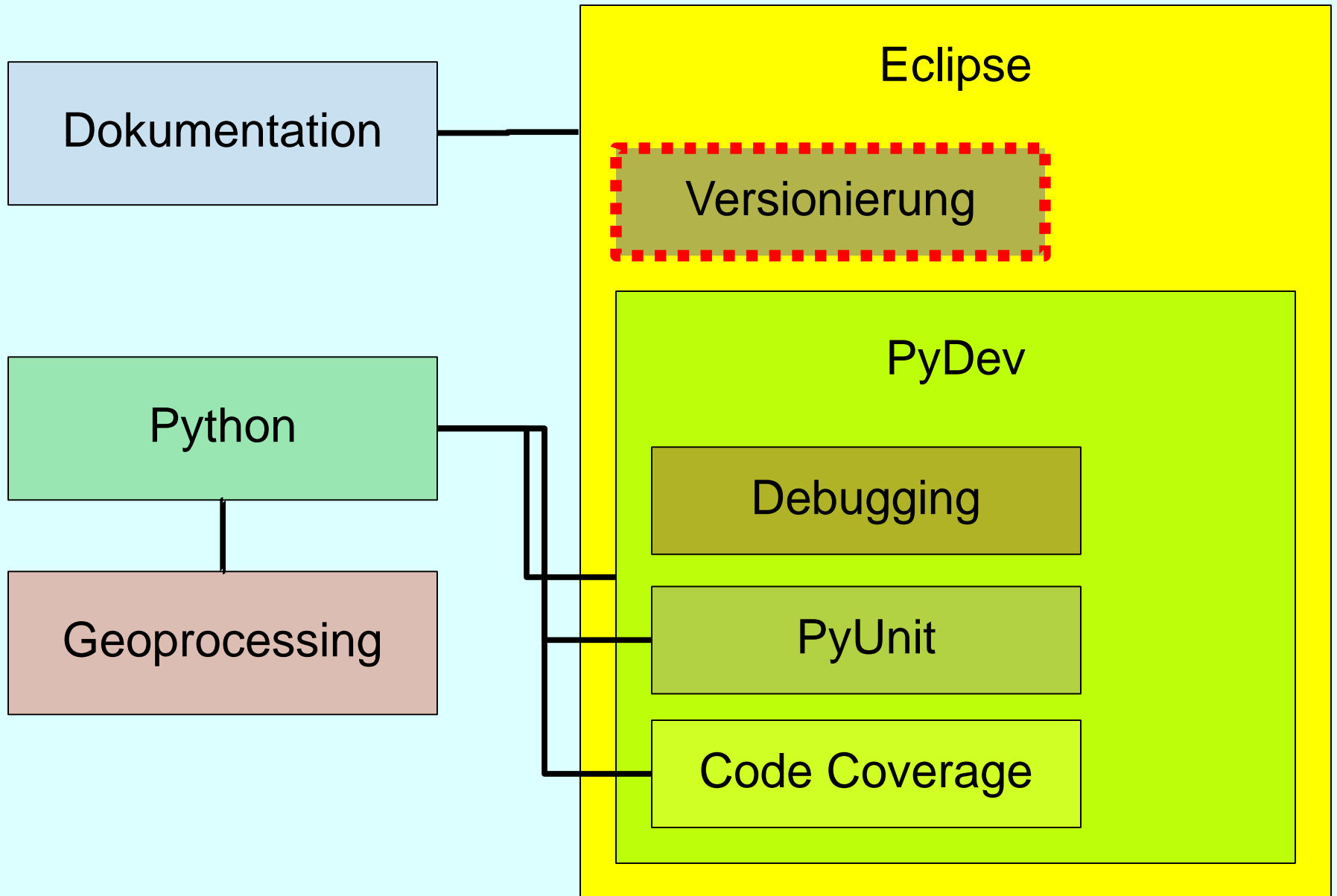
- Suche von nicht getestetem Code bei Unit-Tests
- Auffinden von „totem“ Code

The screenshot shows the PyDev Code Coverage Results View. On the left, a file explorer shows the directory structure of the 'Geoprocessor' module. On the right, a table displays the coverage results for various files.

| Name | Stmts | Exec | Cover | Missing |
|--|-------|------|-------|-------------------------|
| .. c\wgv\Geoprocessor\FeatureHandling.py | 22 | 22 | 100% | |
| .. ssrc\src\wgv\Geoprocessor__init__.py | 0 | 0 | 0% | |
| .. src\wgv\Geoprocessor\TableHandling.py | 146 | 108 | 74% | 45, 50, 55, 60, 65, 124 |
| .. \src\wgv\Geoprocessor\Geoprocessor.py | 13 | 13 | 100% | |
| .. c\wgv\Geoprocessor\TableExceptions.py | 8 | 6 | 75% | 18, 25 |
| TOTAL | 189 | 149 | 78.8% | |

Labels in the image: **Module** (pointing to the file explorer) and **Resultat** (pointing to the table).

Übersichtskarte



Versionierung

Versionierung von Python Modulen

Es existieren Plugins für die gängigen Versionierungsplattformen:

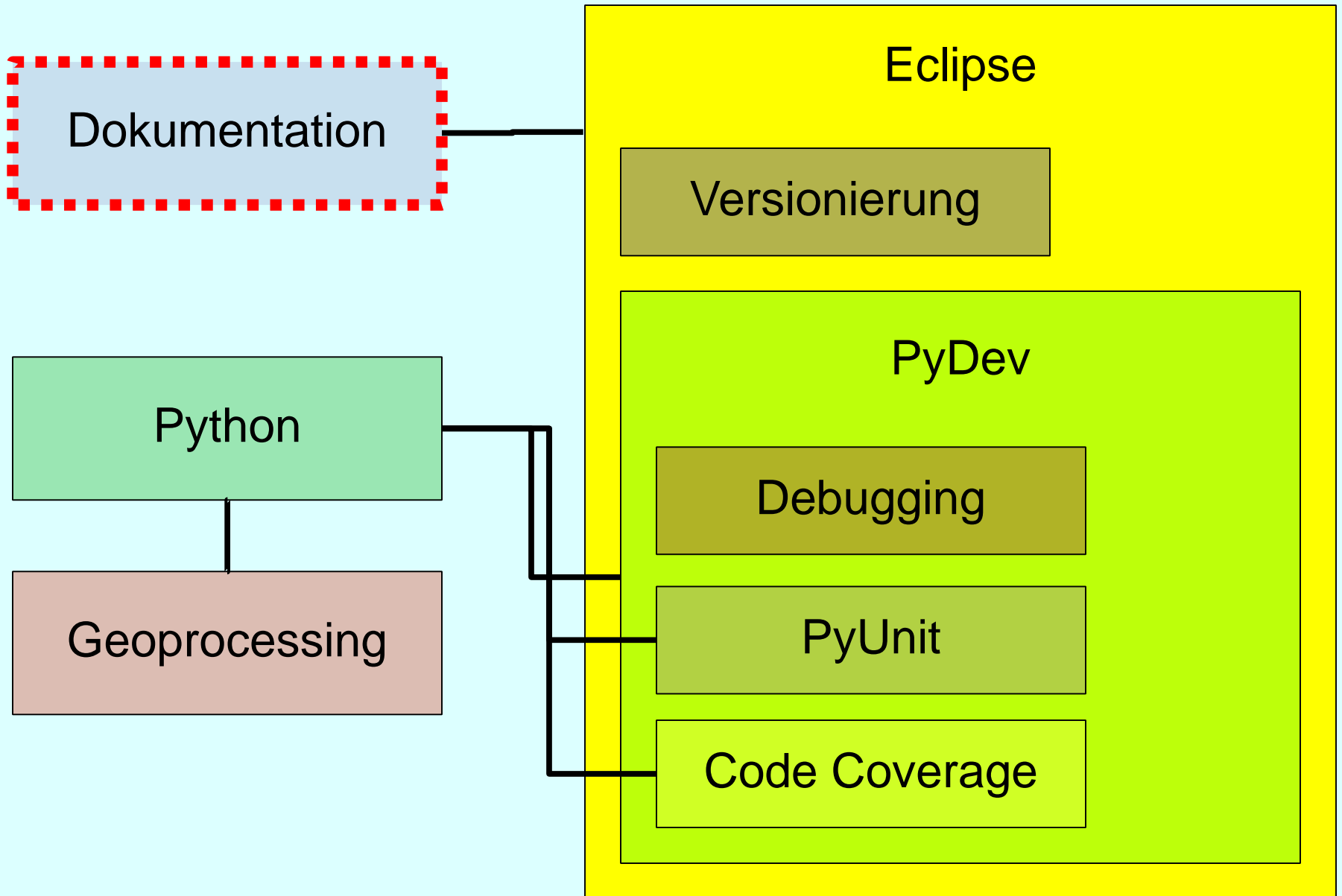
- CVS
- Visual Source Safe
- Subversion

The screenshot shows a version control interface with a commit history table and a detailed view of a specific commit.

| R... | Date | Author | Comment |
|------|------------------|--------|--|
| *45 | 11/13/08 9:58 PM | yvo | - Typenpruefung im Konstruktor der Klasse CTableField |
| 42 | 11/10/08 8:45 PM | yvo | - Vererbungsbaum der einzelnen Klassen korrekt angepasst (keine Mehrfachvererbung) |
| 40 | 11/6/08 11:16 PM | yvo | Kleine Kommentaranpassung |
| 37 | 11/6/08 11:07 PM | yvo | |

| A... | Affected paths | Description |
|------|--------------------------------------|---|
| M | /Python/WGWGeoprocessor/src/wgw/G... | - Typenpruefung im Konstruktor der Klasse CTableField |

Übersichtskarte



Dokumentation

Doxygen auch für Python

Doxygen (www.stack.nl/~dimitri/doxygen/) als externes Tool in Eclipse eingebunden:

- Verwendung von XML-Kommentarsyntax
- Check der Kommentarsyntax, Prüfung Vollständigkeit
- Ausgabe in u.a. in HTML

The screenshot shows the Eclipse IDE interface with several open tabs: FeatureHandlingTest., Parameter.py 37, WGWGeoprocessorTestS, TableHandling.py, and Geoprocessor. The main editor displays the source code documentation for the Geoprocessor class, generated by Doxygen. The documentation is titled "Source-Code Dokumentation" and includes a navigation menu with buttons for "Hauptseite", "Namensbereiche", "Klassen", "Dateien", "Aufzählung der Klassen", "Klassenhierarchie", and "Klassen-Elemente". The main content area is titled "Geoprocessor Aufzählung der Klassen" and contains the text: "Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:". Below this, there is a table with two columns. The first column contains the class name: "wgw::Geoprocessor::FeatureHandling::CFeatureHandling". The second column contains a description: "Von der Klasse CGeoprocessor abgeleitete Subklasse. Die Klasse dient der Bearbeitung von". On the right side of the IDE, there is a vertical toolbar with buttons for "createT", "deleteT", and "getNum", and a small code editor showing "m>".

Eclipse mit PyDev

bildet eine geeignete Plattform für die Entwicklung für Geoprocessing.



- Unterstützung der Sprache Python
- Hilfreich bei grösseren Projekten
(Packages, Objektorientierte Lösungen, ...)
- Zahlreiche Hilfsmittel moderner IDE's
(Unit-Test, Coverage, Refactoring, ...)



- Overhead bei kleinen Projekten /Skripten
- Grosser Memory-Verbrauch von Eclipse
- Schwerfällig bei „Alltagsproblemen“
(Console)

Schluss

TIERPATENSCHAFTEN

TIERPARK DÄHLHÖLZLI

TIERPARKVEREIN



BERN



URKUNDE



WaterGisWeb AG

Donnerbühlweg 41, 3012 Bern

ist ab

Dienstag, 1. Januar 2008

Pate eines Grünen Baumpythons des
Tierparks Dählhölzli Bern