

F. Deißeböck, B. Hummel, E. Jürgens, T. Seifert

Kontinuierliche Qualitätsanalyse mit ConQAT

25. April 2007
Perlen der Informatik



Q-Modell »Wartbarkeit«

- Beschreibung der Einflussfaktoren
- Erklärung der Zusammenhänge
- Anleitung zur Bewertung
- ▶ Einflussfaktoren sind zahlreich und vielfältig

Q-Prozess

- »Erhaltung ist einfacher als Reparatur«
- zeitnahes/kontinuierliches Qualitäts-Controlling
- Erhöhung der Disziplin
- ▶ höhere Produktqualität und geringere Wartungsaufwände

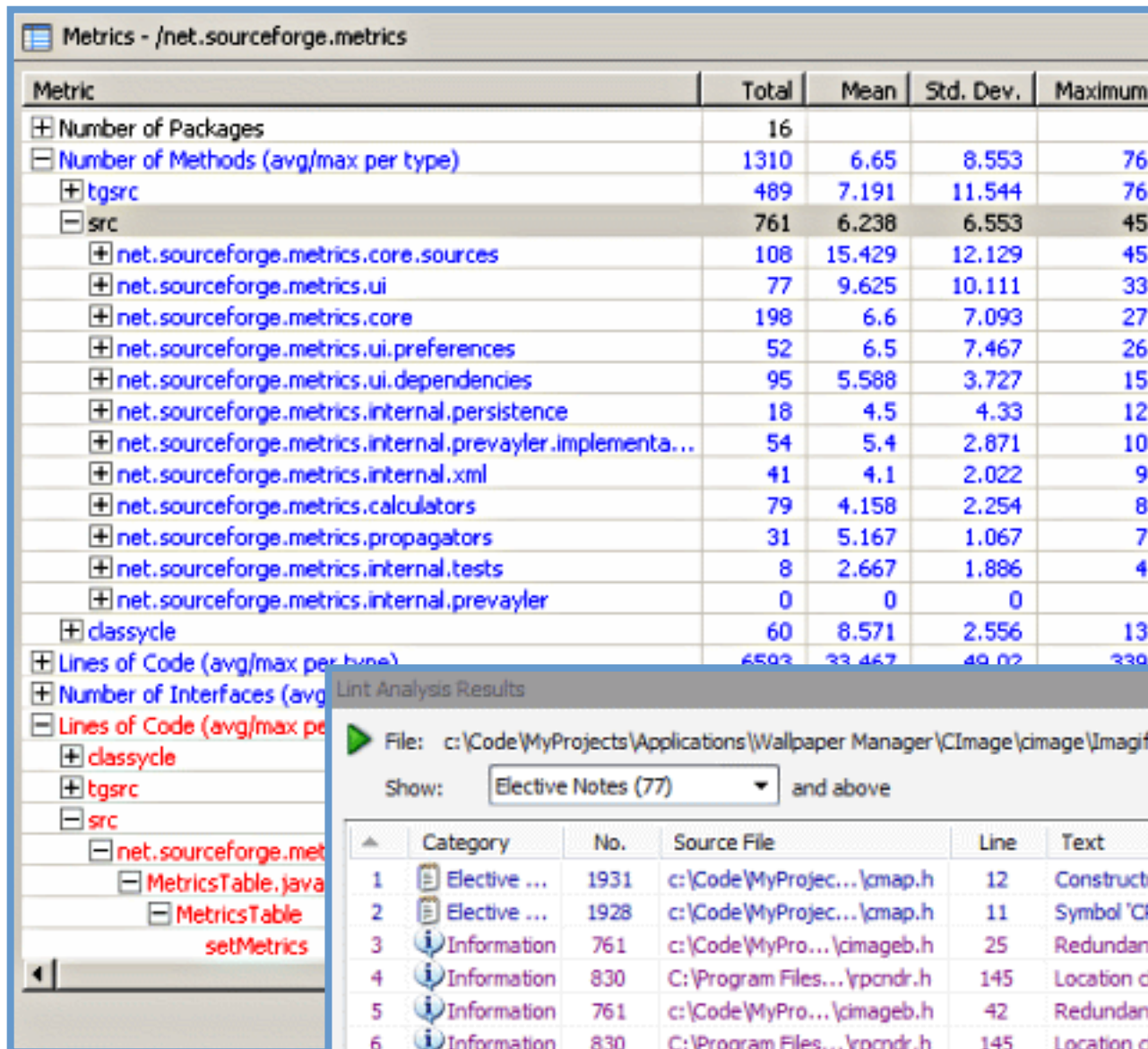
Tool-Support

- Qualitäts-Controlling ist teuer
- Automatisierung notwendig
- ▶ Entwicklung von ConQAT

Anforderungen

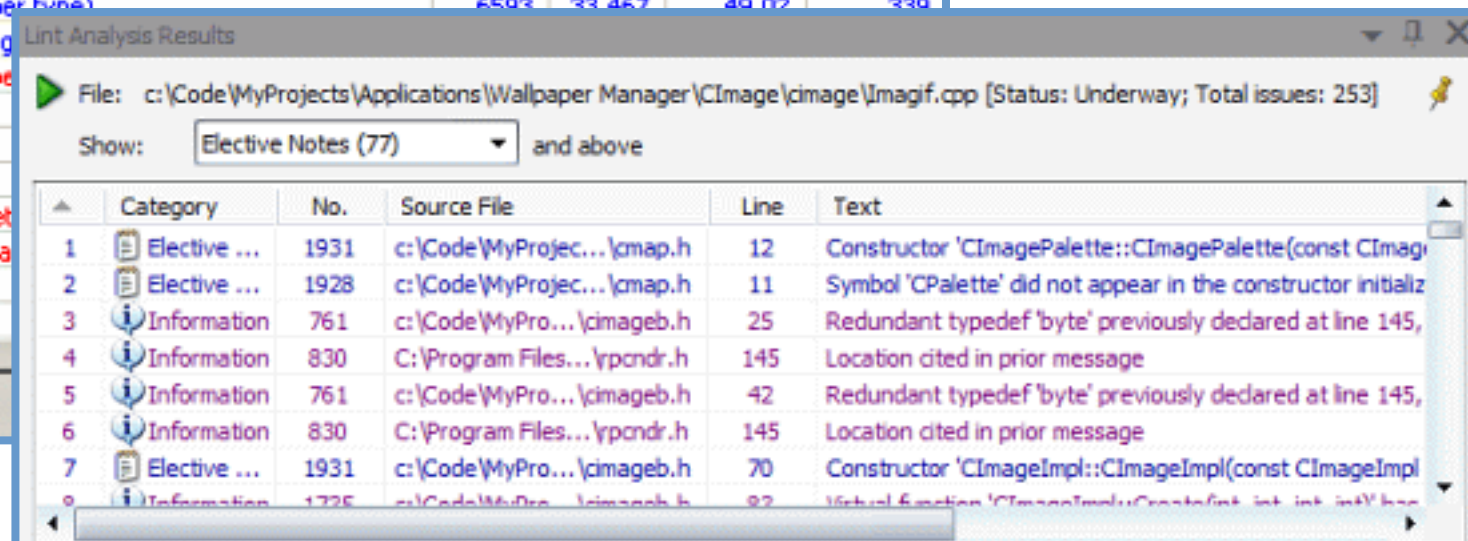
- Integrative Darstellung
- Batch-Betrieb
- Informations-Verdichtung
- Vielfältigkeit
- Erweiterbarkeit
- Flexibilität
- Skalierbarkeit

Metrik-Werkzeuge



Metrics - /net.sourceforge.metrics

Metric	Total	Mean	Std. Dev.	Maximum
+ Number of Packages	16			
- Number of Methods (avg/max per type)	1310	6.65	8.553	76
+ tgsrc	489	7.191	11.544	76
- src	761	6.238	6.553	45
+ net.sourceforge.metrics.core.sources	108	15.429	12.129	45
+ net.sourceforge.metrics.ui	77	9.625	10.111	33
+ net.sourceforge.metrics.core	198	6.6	7.093	27
+ net.sourceforge.metrics.ui.preferences	52	6.5	7.467	26
+ net.sourceforge.metrics.ui.dependencies	95	5.588	3.727	15
+ net.sourceforge.metrics.internal.persistence	18	4.5	4.33	12
+ net.sourceforge.metrics.internal.prevayler.implementa...	54	5.4	2.871	10
+ net.sourceforge.metrics.internal.xml	41	4.1	2.022	9
+ net.sourceforge.metrics.calculators	79	4.158	2.254	8
+ net.sourceforge.metrics.propagators	31	5.167	1.067	7
+ net.sourceforge.metrics.internal.tests	8	2.667	1.886	4
+ net.sourceforge.metrics.internal.prevayler	0	0	0	
+ classycle	60	8.571	2.556	13
+ Lines of Code (avg/max per type)	6593	33.467	49.02	339
+ Number of Interfaces (avg/max per type)				
- Lines of Code (avg/max per type)				
+ classycle				
+ tgsrc				
- src				
- net.sourceforge.metrics				
- MetricsTable.java				
- MetricsTable				
- setMetrics				



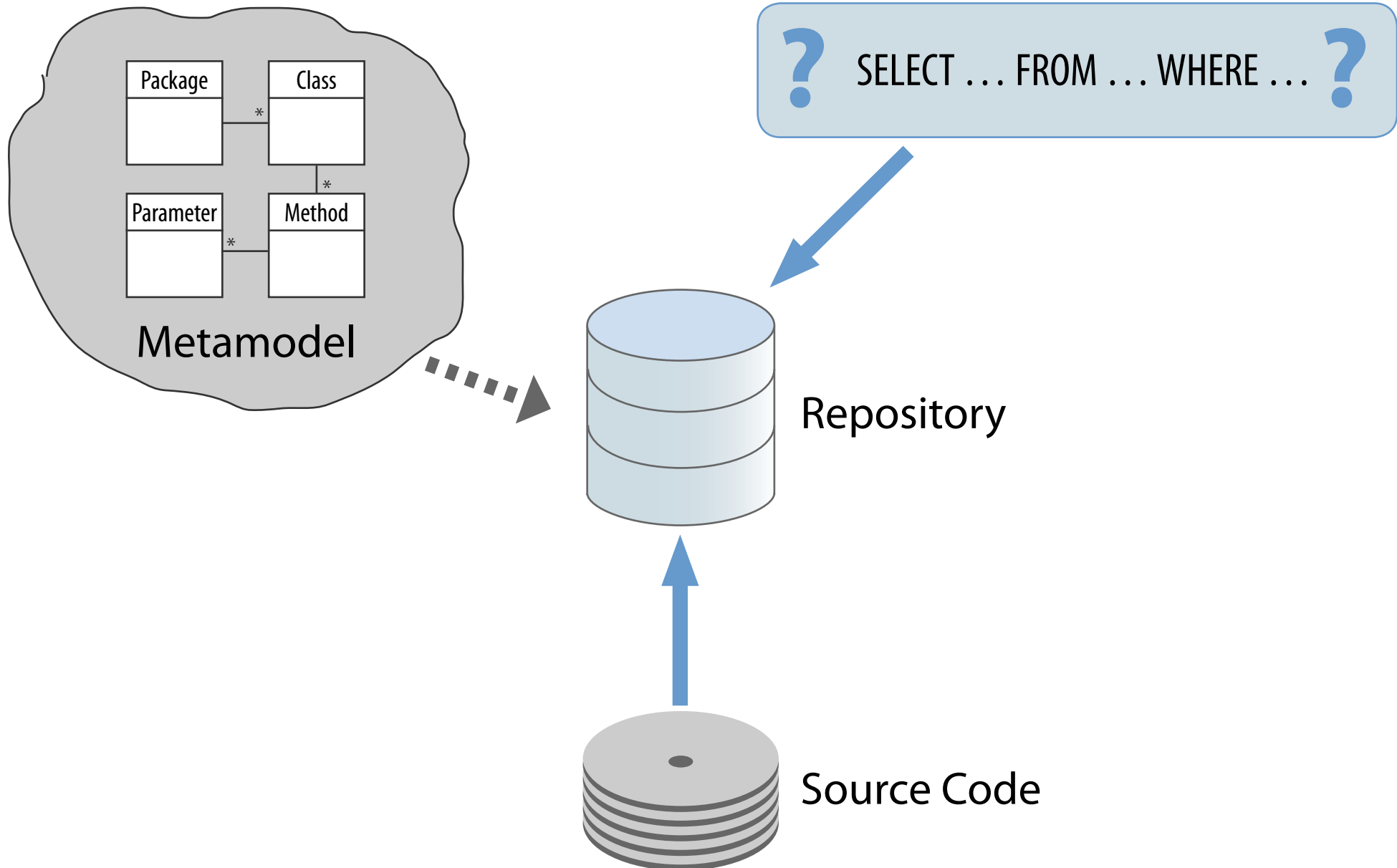
Lint Analysis Results

File: c:\Code\MyProjects\Applications\Wallpaper Manager\CImage\cimage\Imagif.cpp [Status: Underway; Total issues: 253]

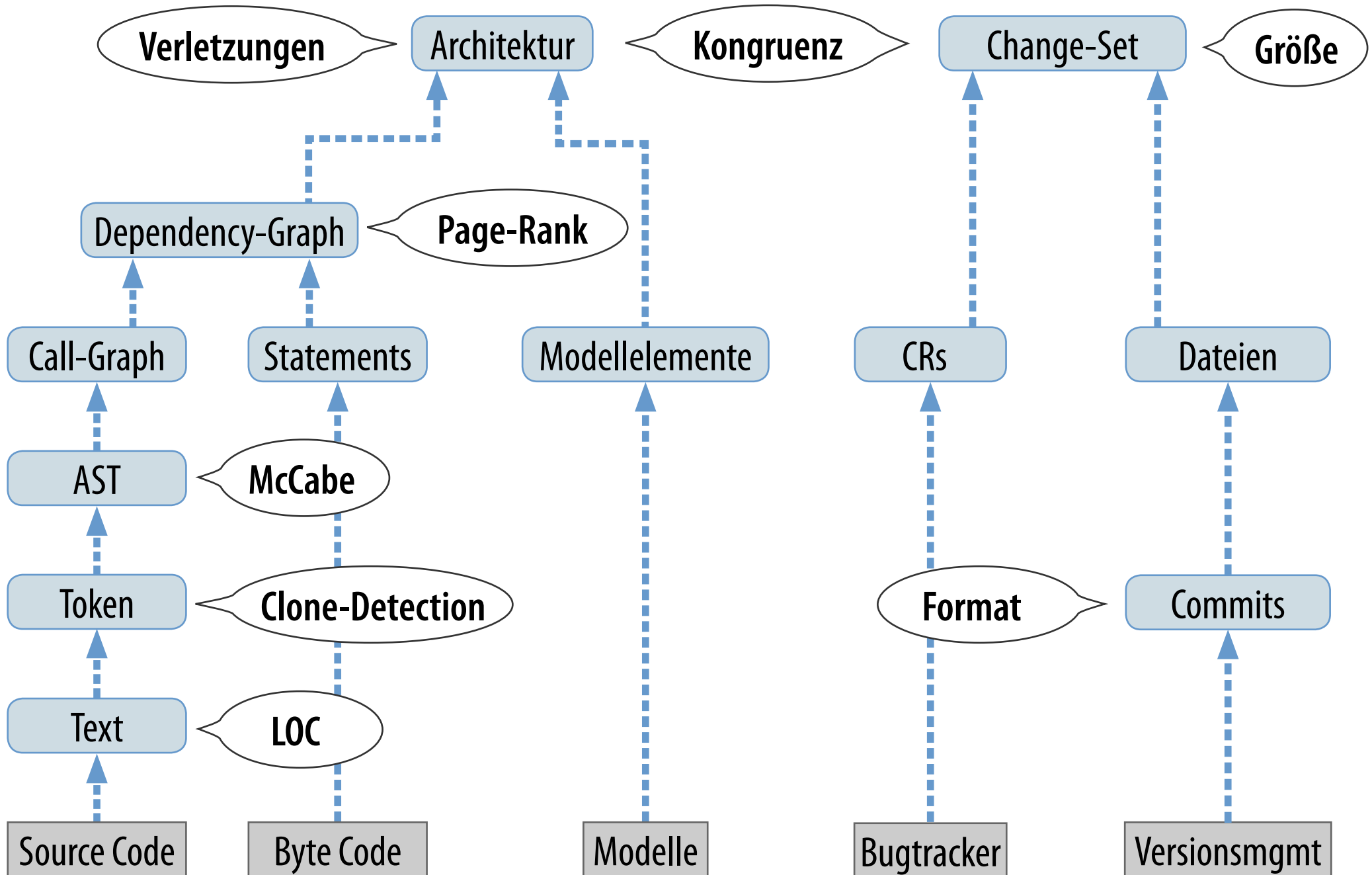
Show: Elective Notes (77) and above

	Category	No.	Source File	Line	Text
1	Elective ...	1931	c:\Code\MyProjec... \cmap.h	12	Constructor 'CImagePalette::CImagePalette(const CImage...
2	Elective ...	1928	c:\Code\MyProjec... \cmap.h	11	Symbol 'CPalette' did not appear in the constructor initialz
3	Information	761	c:\Code\MyPro... \cimageb.h	25	Redundant typedef 'byte' previously declared at line 145,
4	Information	830	C:\Program Files... \ypcndr.h	145	Location cited in prior message
5	Information	761	c:\Code\MyPro... \cimageb.h	42	Redundant typedef 'byte' previously declared at line 145,
6	Information	830	C:\Program Files... \ypcndr.h	145	Location cited in prior message
7	Elective ...	1931	c:\Code\MyPro... \cimageb.h	70	Constructor 'CImageImpl::CImageImpl(const CImageImpl
8	Information	1735	c:\Code\MyPro... \cimageb.h	82	Virtual function 'CImageImpl::Create(int, int, int)' has

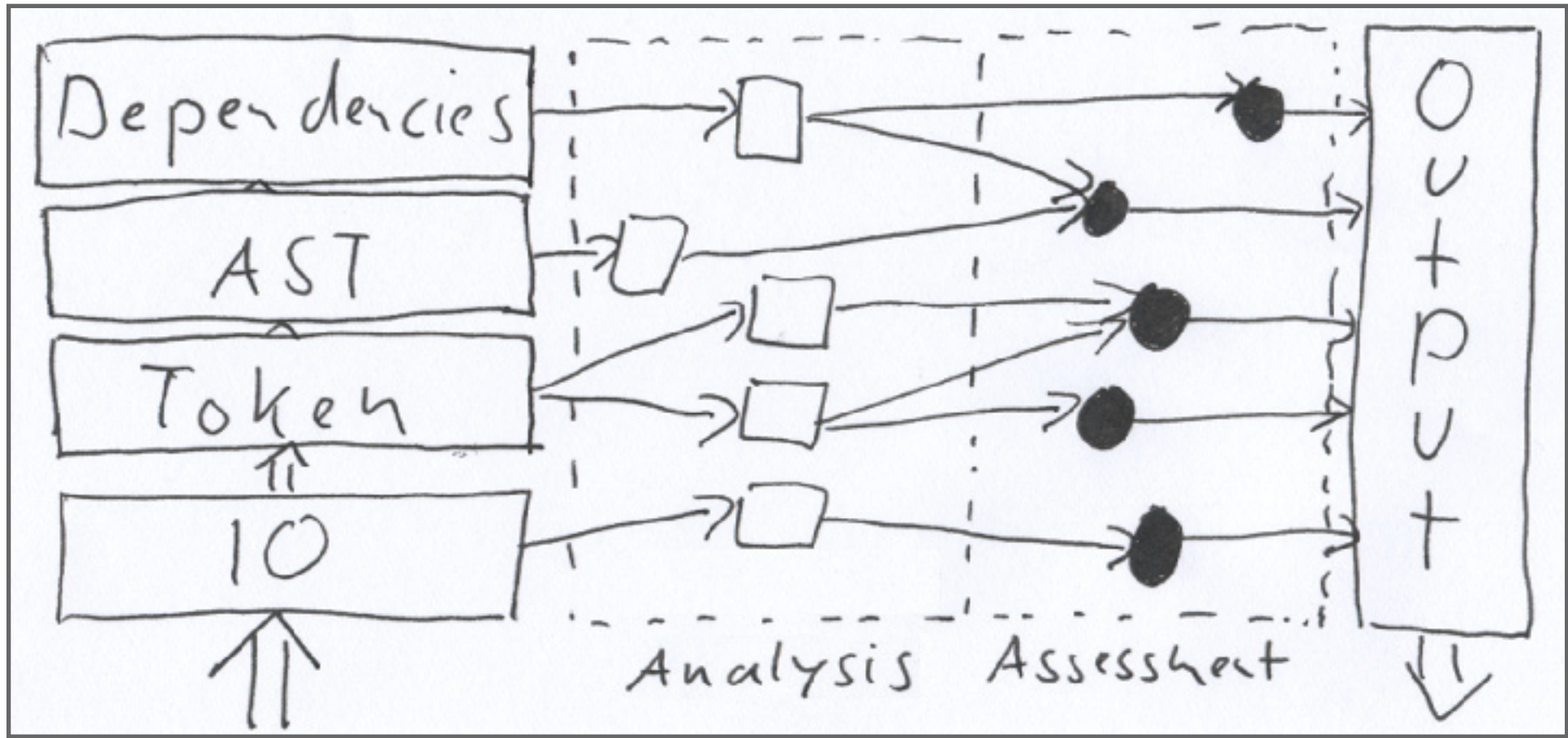
Repository-Based Tools



Realität



Architektur-Entwurf

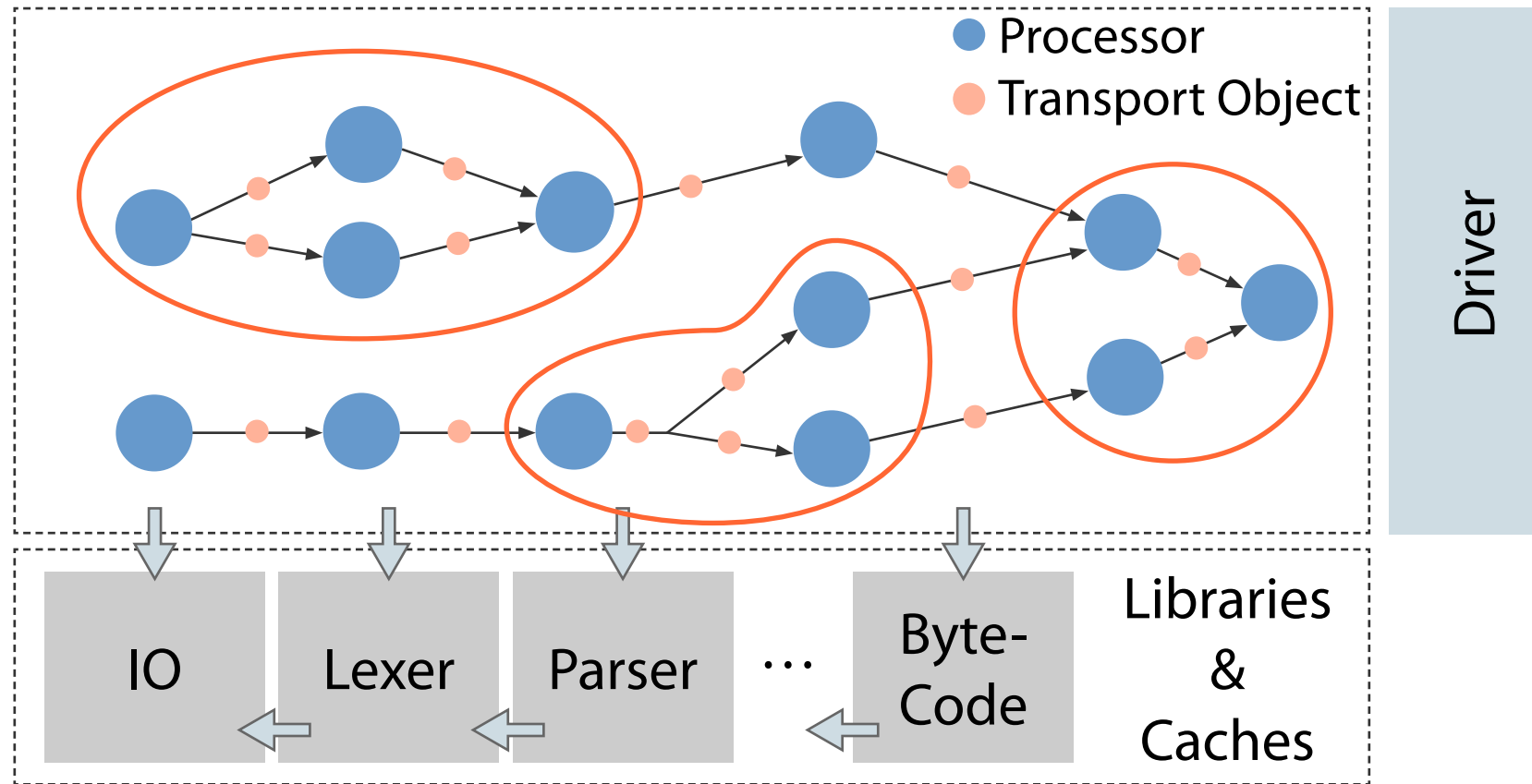


Prototypen

- Entwicklung verschiedener Prototypen
- Spektrum
 - Flexibilität vs Mächtigkeit
 - Komposition
 - Infrastruktur
- keine »ideale« Stelle auf dem Spektrum
- ▶ Ausbruch aus dem Spektrum

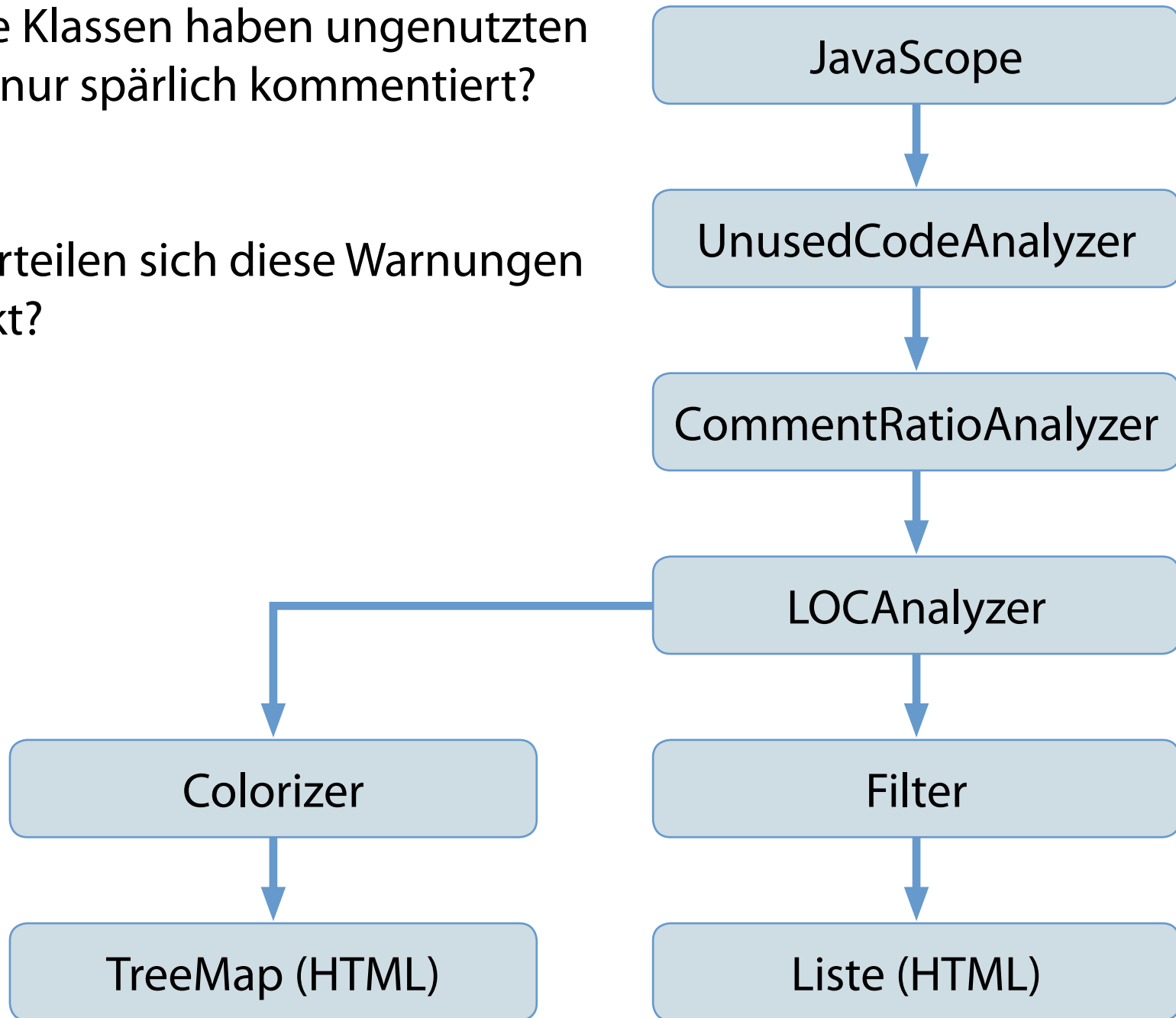


Architektur



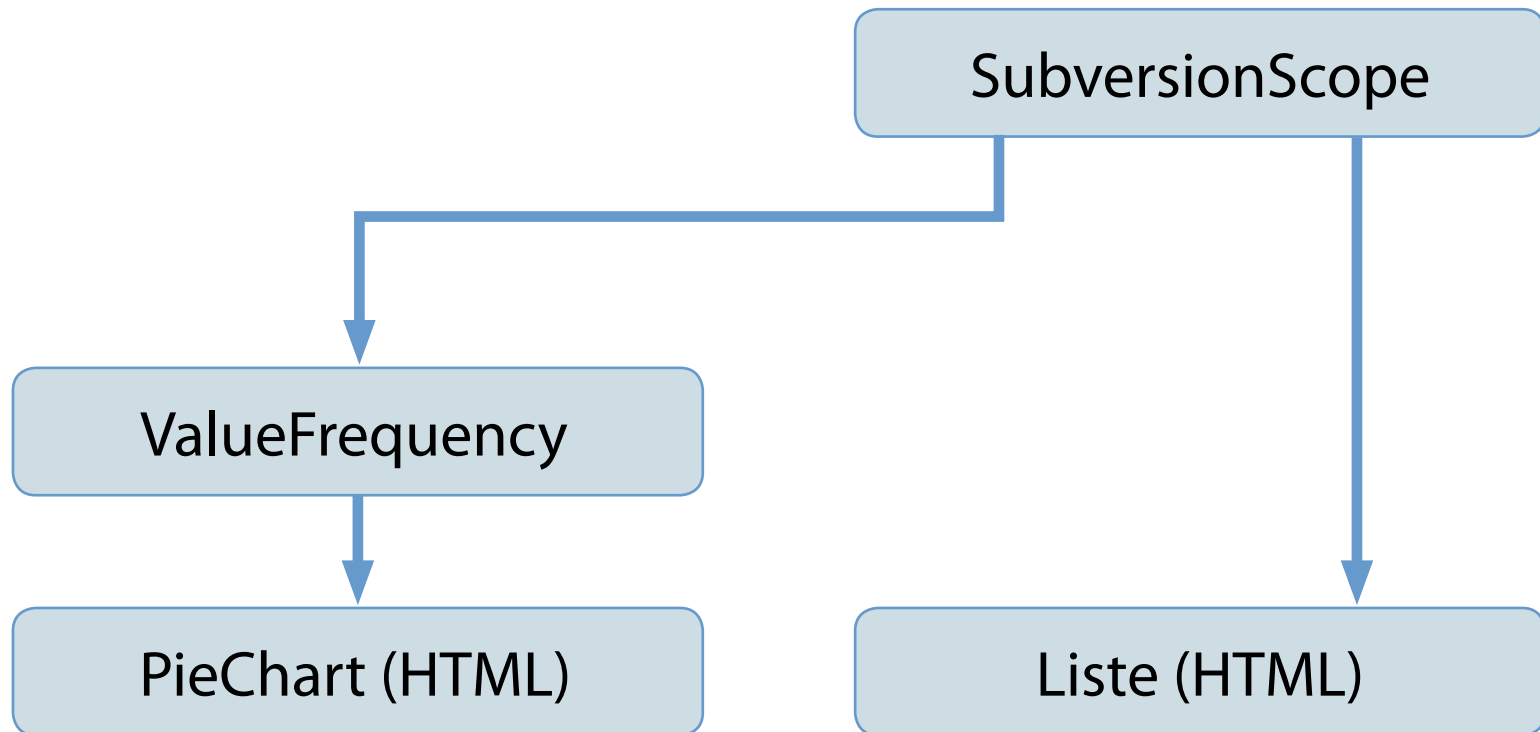
Beispiel I

- Frage 1: Welche Klassen haben ungenutzten Code und sind nur spärlich kommentiert?
- Frage 2: Wie verteilen sich diese Warnungen über das Projekt?



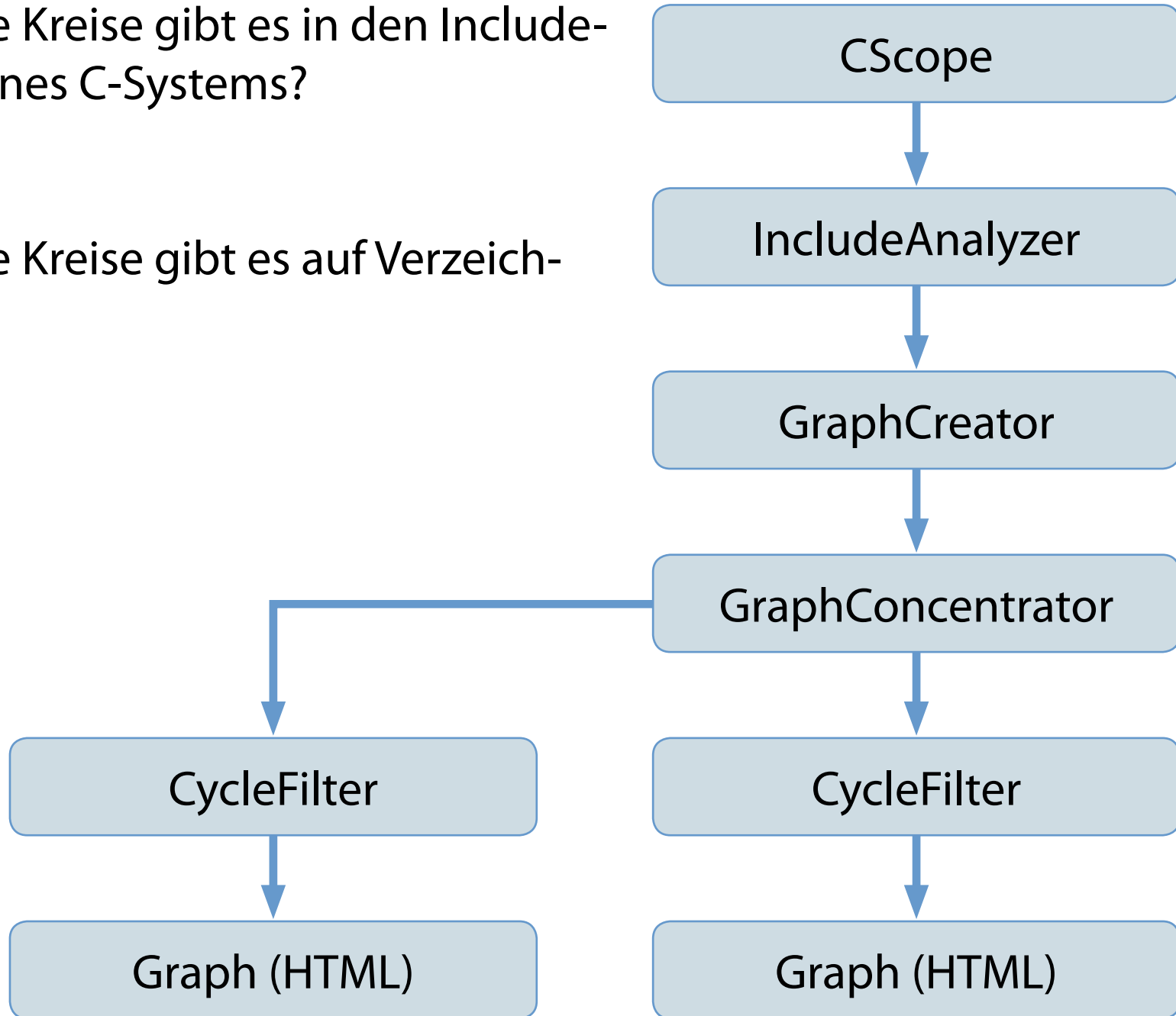
Beispiel II

- Frage 1: Welche Commits gab es in der letzten Woche?
- Frage 2: Wie verteilen sich die Commits bzgl. der Autorenschaft?

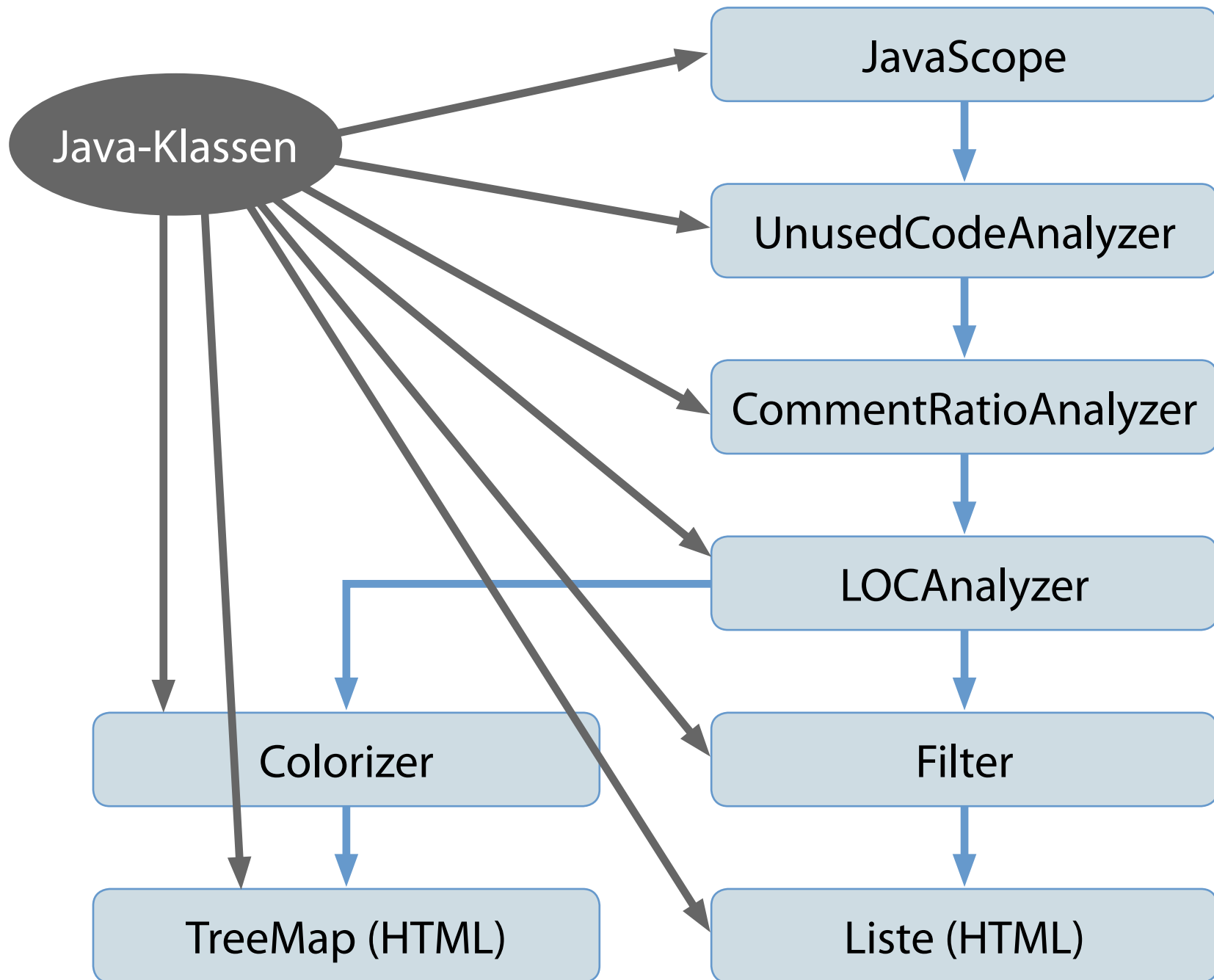


Beispiel III

- Frage 1: Welche Kreise gibt es in den Include-Strukturen meines C-Systems?
- Frage 2: Welche Kreise gibt es auf Verzeichnisebene?



Implementierung



Konfiguration

```
<processor id="source" class="FileSystemScope">  
  <root dir="src" />  
</processor>
```

```
@AConfigProcessor(description = "File systems scope")  
public class FileSystemScope implements IProcessor {  
  
  @AConfigElement(name = "root")  
  public void setRootDirectory(  
    @AConfigAttribute(name = "dir") String rootDir) {  
    ...  
  }  
  
  public IFileSystemElement process() throws ConQATException {  
    ...  
  }  
}
```

```
<processor id="loc-analysis" class="LOCAalyzer">  
  <input ref="@source" />  
</processor>
```

```
@AConfigProcessor(description = „Count LOC“)  
public class LOCAalyzer implements IProcessor {  
  
  @AConfigElement(name = „input“)  
  public void setRoot(  
    @AConfigAttribute(name = „ref“) IFileSystemElement root) {  
    ...  
  }  
  ...  
}
```

Einsatzgebiete

Artefakte

- Java
- C#
- C, C++, Cobol, PL/1
- Simulink, Stateflow
- Text-Dateien
- Unit-Tests
- Bugzilla
- Subversion

Analysen

- Architekturanalyse
- Clone-Detection
- Metriken
- Suche
- Integration von
 - PMD
 - FindBugs
 - CodeInspector
 - Klocwork
 - ...

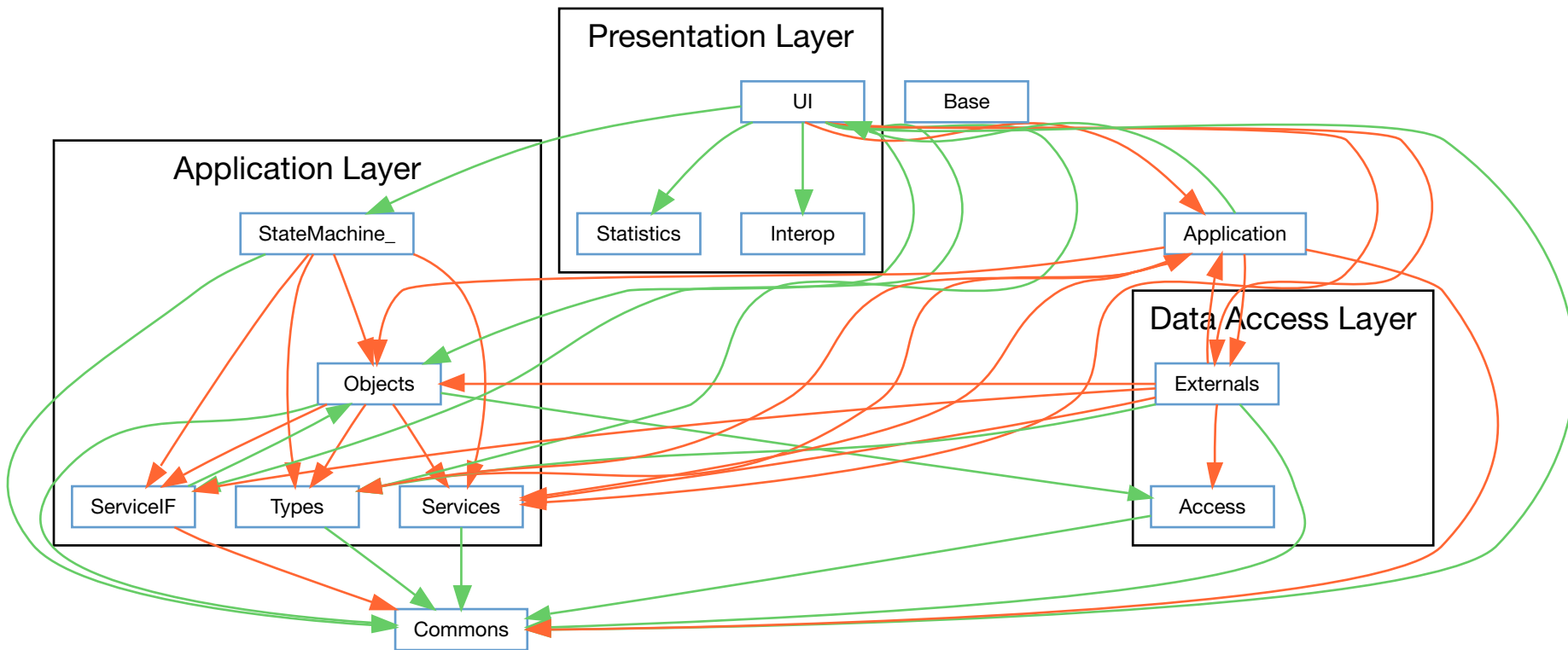
Ausgabe

- Tabellen
- Text
- Diagramme
 - Linien
 - Kreis
 - Radar
- Graphen
- Tree-Maps

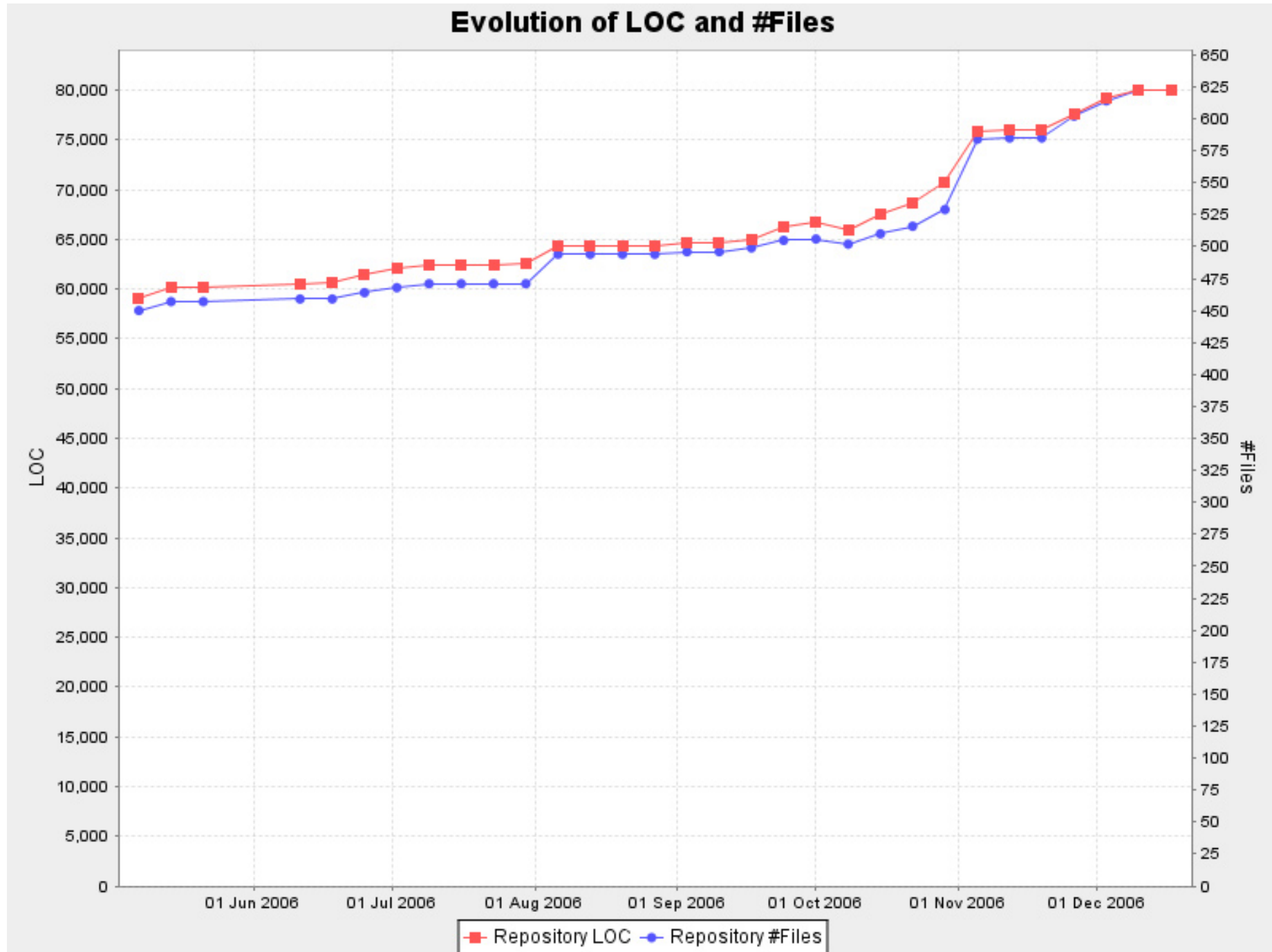
Demo

- Aggregation
- Tree-Maps
- Radar-Plots
- Graphen
- Graph-Metriken
- Architektur
- Trendanalyse

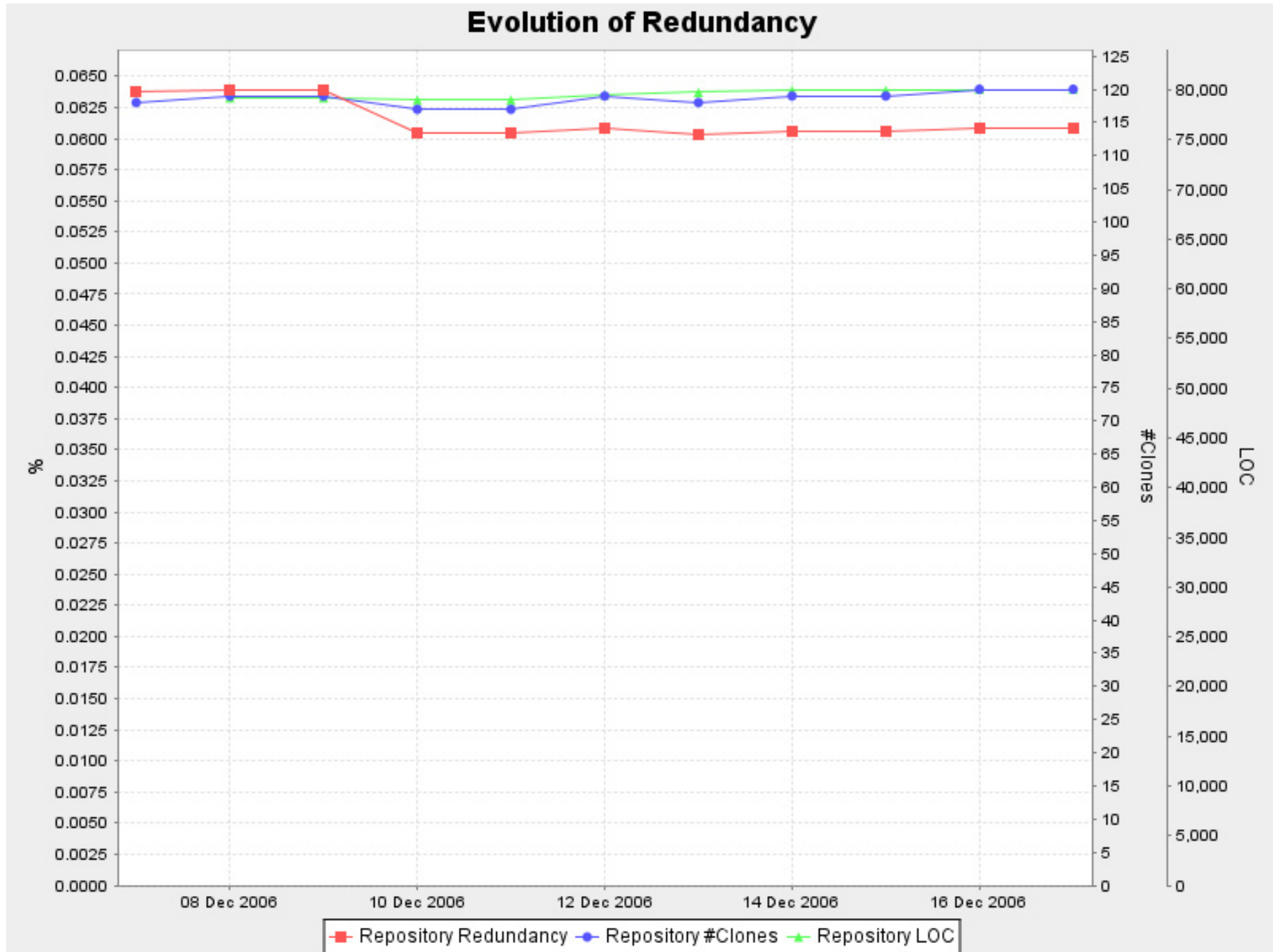
Architektur-Überprüfung



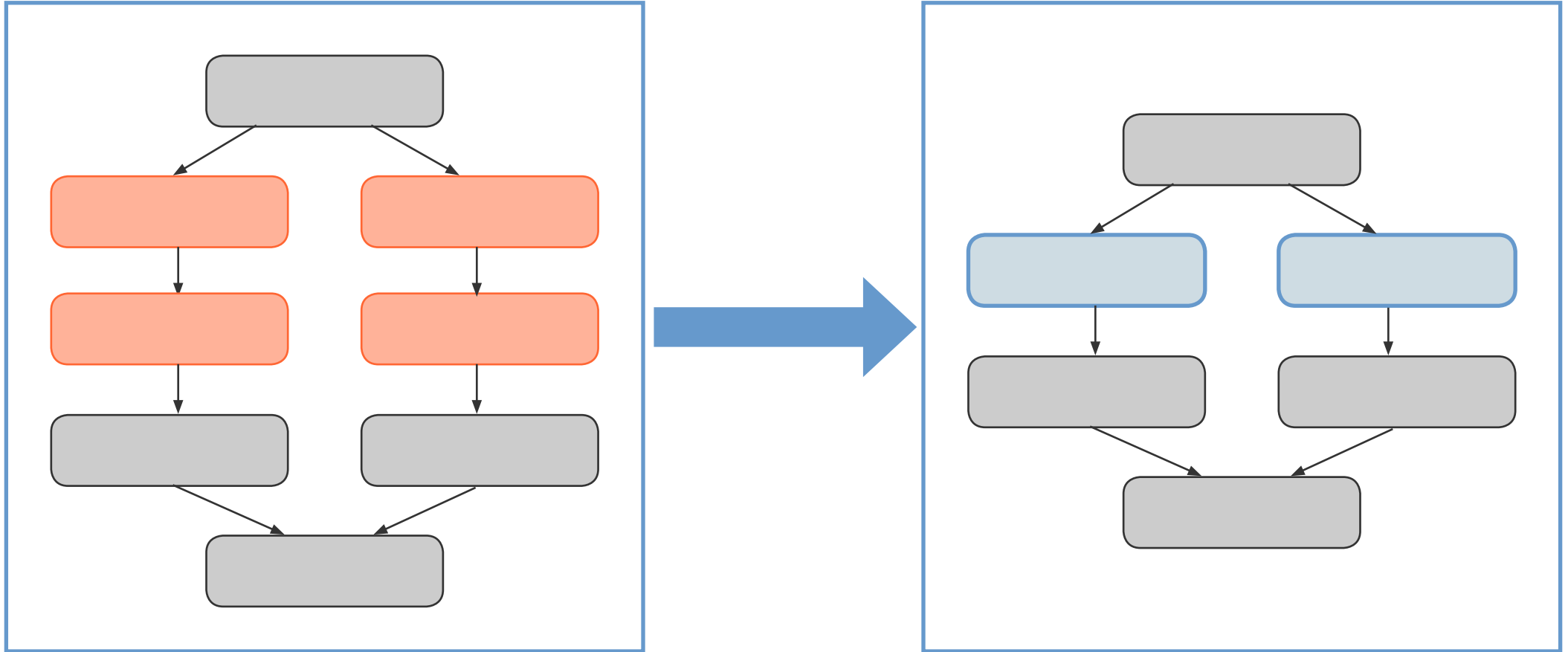
Trendanalyse I



Trendanalyse II



Kompositions-Mechanismus



Weitere Features

- Bundle-Mechanismus (Plug-Ins)
- ConQATDoc
- Selbstdiagnose
- Build-Mechanismus

- Morgen: Pre-Alpha-Release
- Bald: Beta-Release
- Mai: erster offizieller Release (Open Source)
- Langfristig: Etablierung als *das* Open-Source-Analyseframework ;-)