

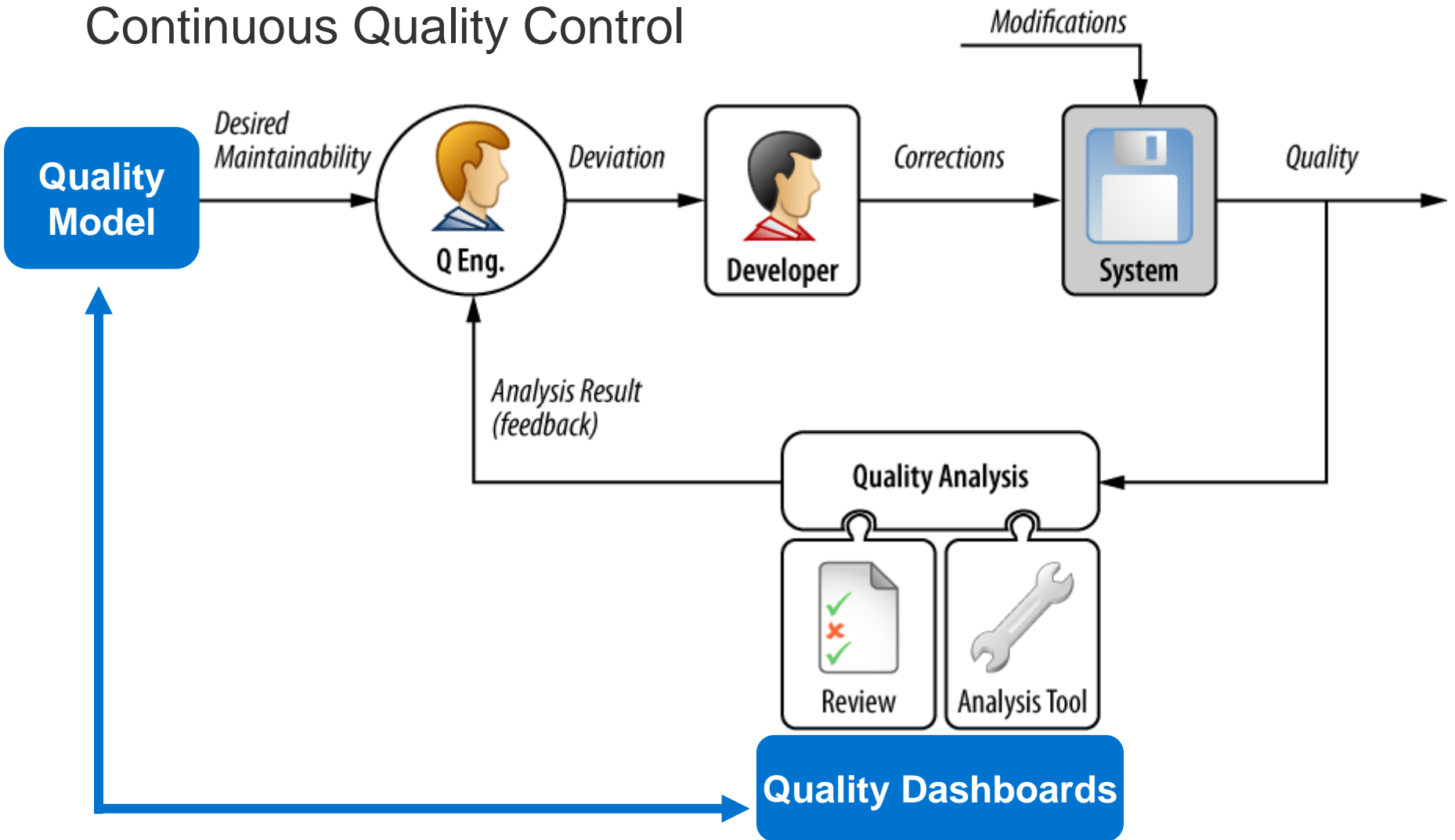
Tool-supported  
**Software**  
**product quality**  
**control**

**Dr. Stefan Wagner**  
**Dr. Florian Deißeböck**  
Technische Universität München

Google Developer Day

Munich  
November 9, 2010

# Continuous Quality Control

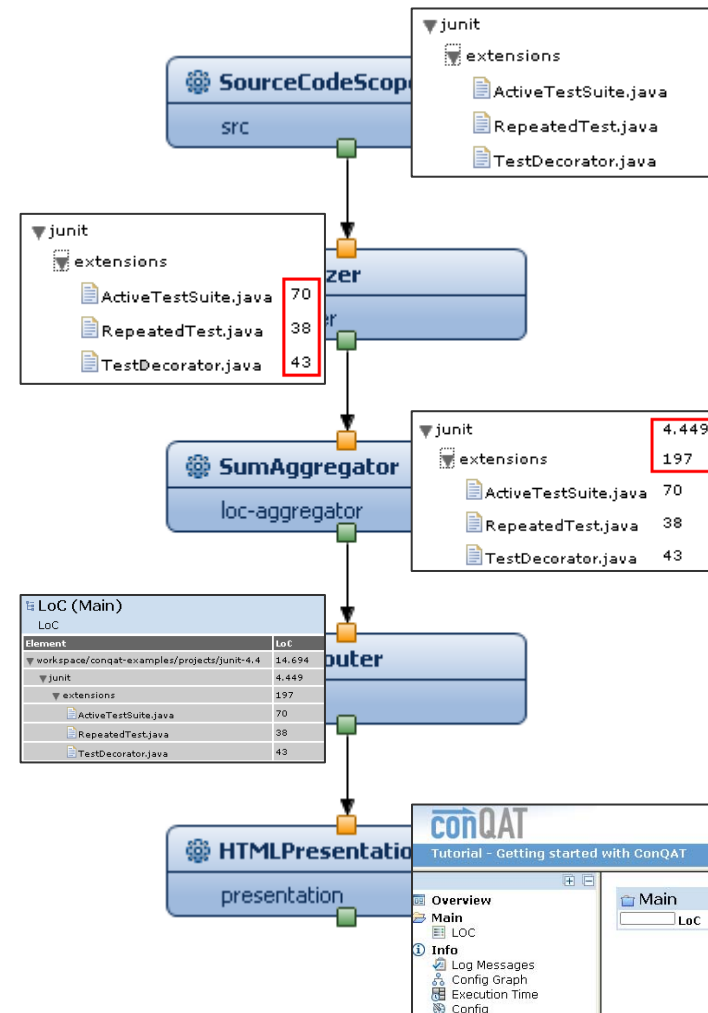


# Challenges

- **Customizability, Extensibility**  
Allow tailoring to project-specific needs for precision and relevance
- **Aggregation and Visualization**  
Minimize Time for result comprehension
- **Trend Analysis**  
Capture development over time and make improvements tangible
- **Diversity of analyzed artifacts**  
Allow analysis of relevant artifacts beyond code (Models, DB, ...)
- **Autonomous operation, Performance, Scalability**  
Allow regular, automated execution for timely result data

# ConQAT

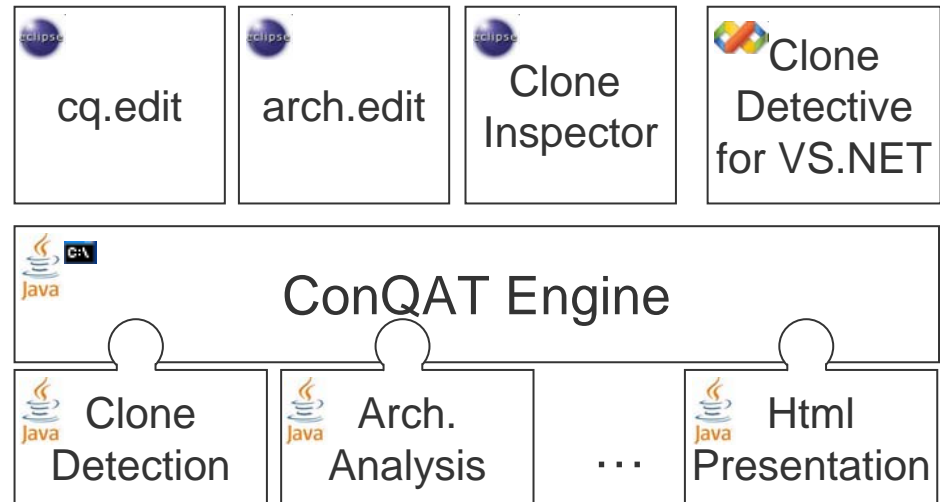
- **Graphical DSL** to specify analysis configuration
- **Processors** perform analysis operations
- **Edges** describe data flow between processors
- **Blocks** make configuration fragments reusable



# ConQAT

## ConQAT Engine

- Executes analysis configurations
- Performs Type Checking
- Performs Caching (Files, Tokens, Ast Fragments, ...)
- Runs in Batch mode



## ConQAT Bundles (=Plugins)

- Implement actual analyses as libraries of processors and blocks
- Can be developed and deployed independently of ConQAT

# Architecture Conformance Analysis

## Goal: Discover

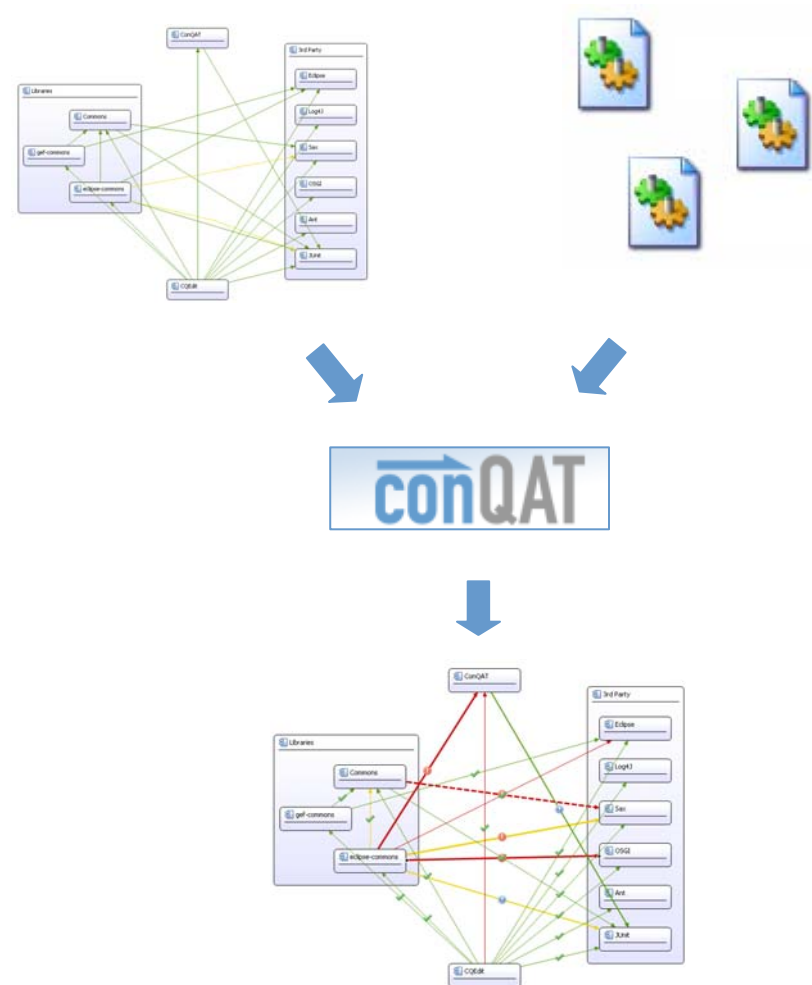
- Architecture violations
- Gaps in Arch. Doc.

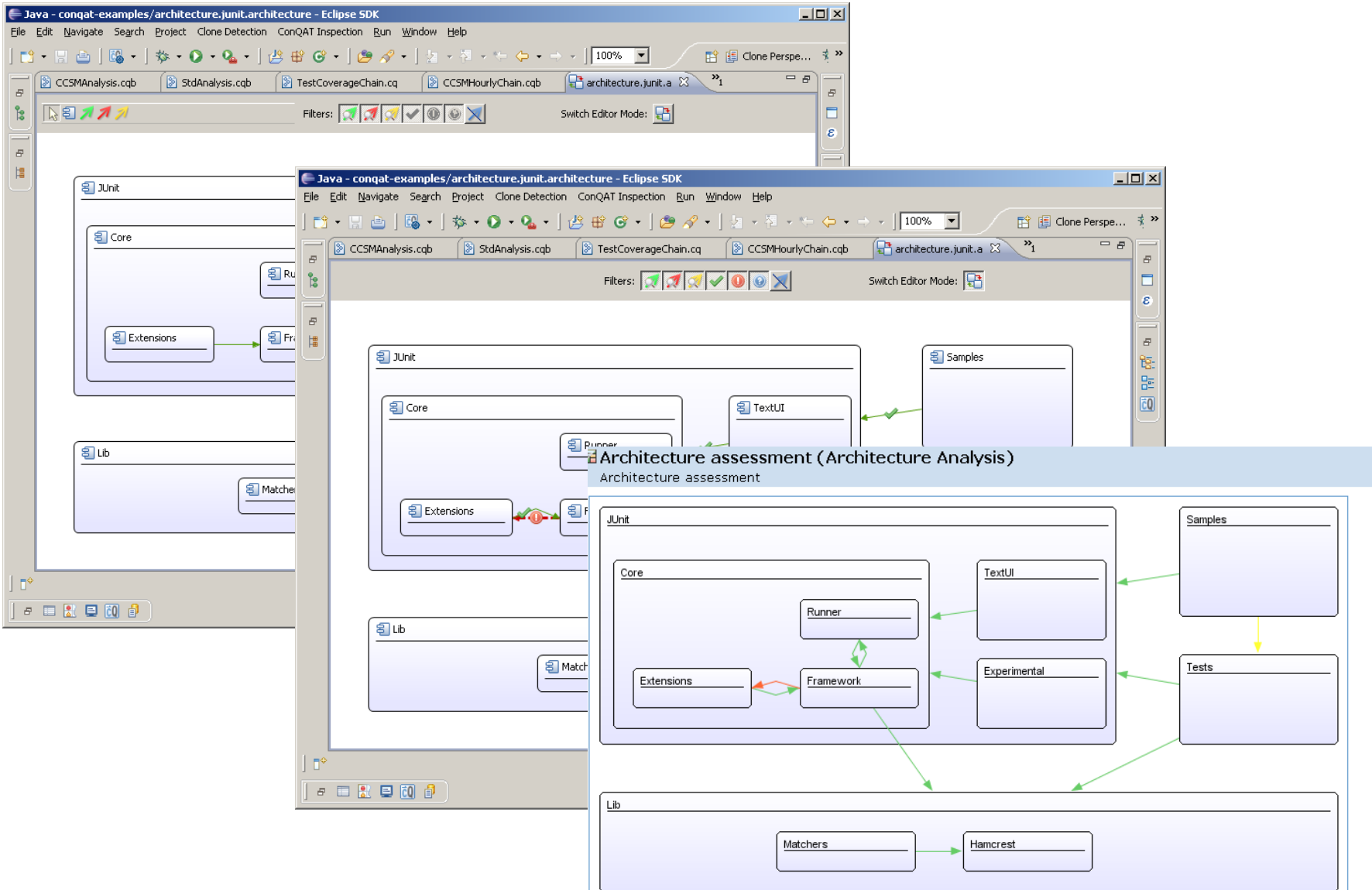
## Motivation

- Program Comprehension
- Impact-Analysis

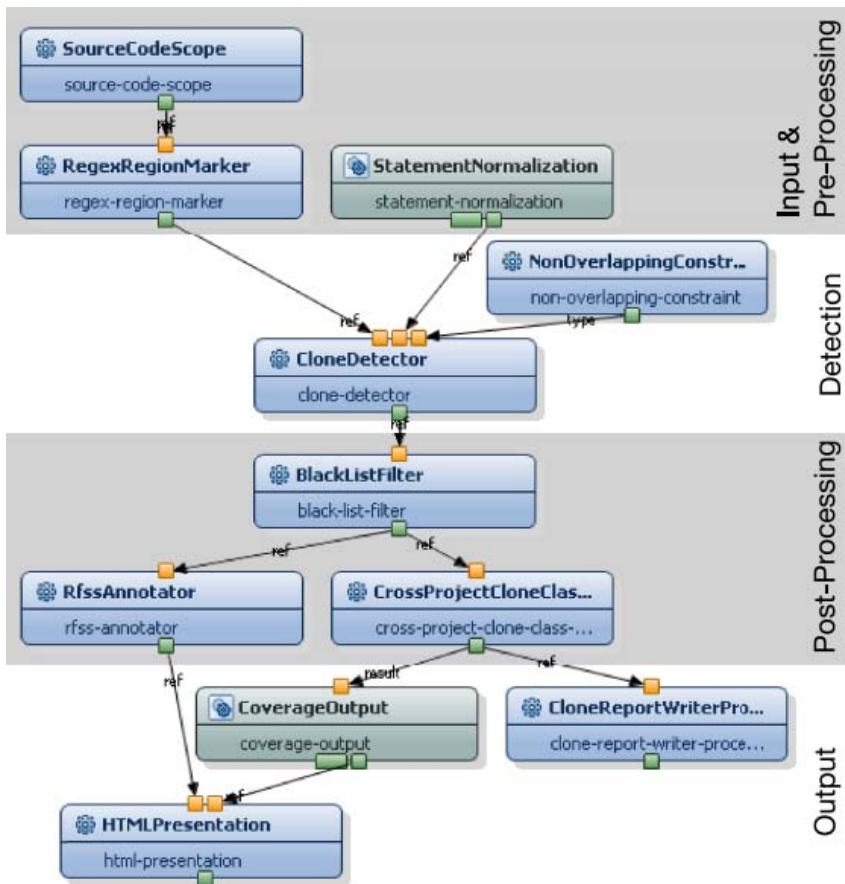
## Approach

- Documentation as development artifact
- Automated Conformance analysis





# Tailorable Clone Detection



- Java, C#, C/C++, VB, Cobol, PL/I, Words or Lines

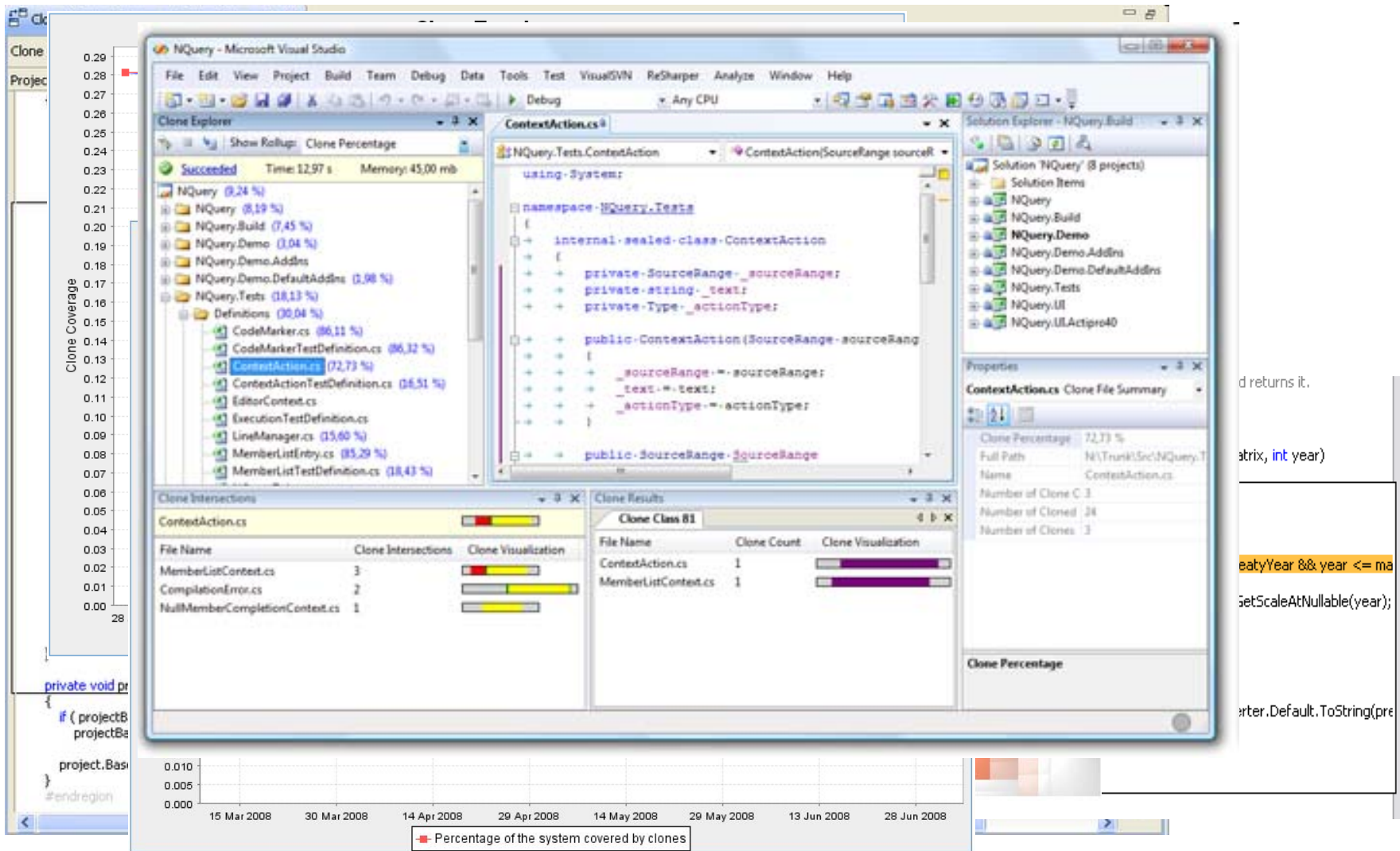
- Shapers

- Ungapped Clones
- Gapped Clones

- Filtering, Blacklisting
- Metric Computation

- XML Report
- Trends
- Dashboard (+Visualizations)





d returns it.

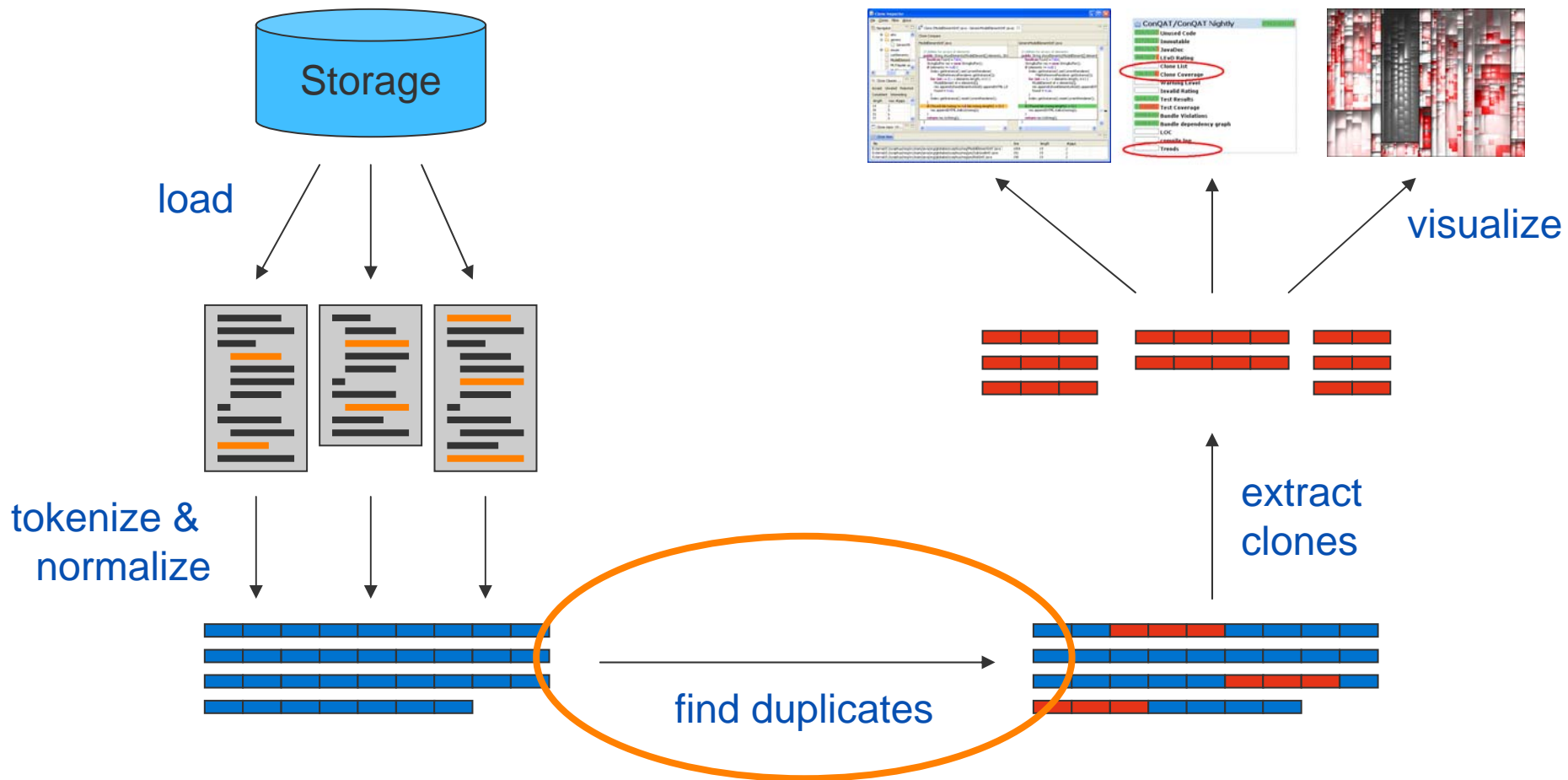
atrix, int year)

eatyYear && year <= ma

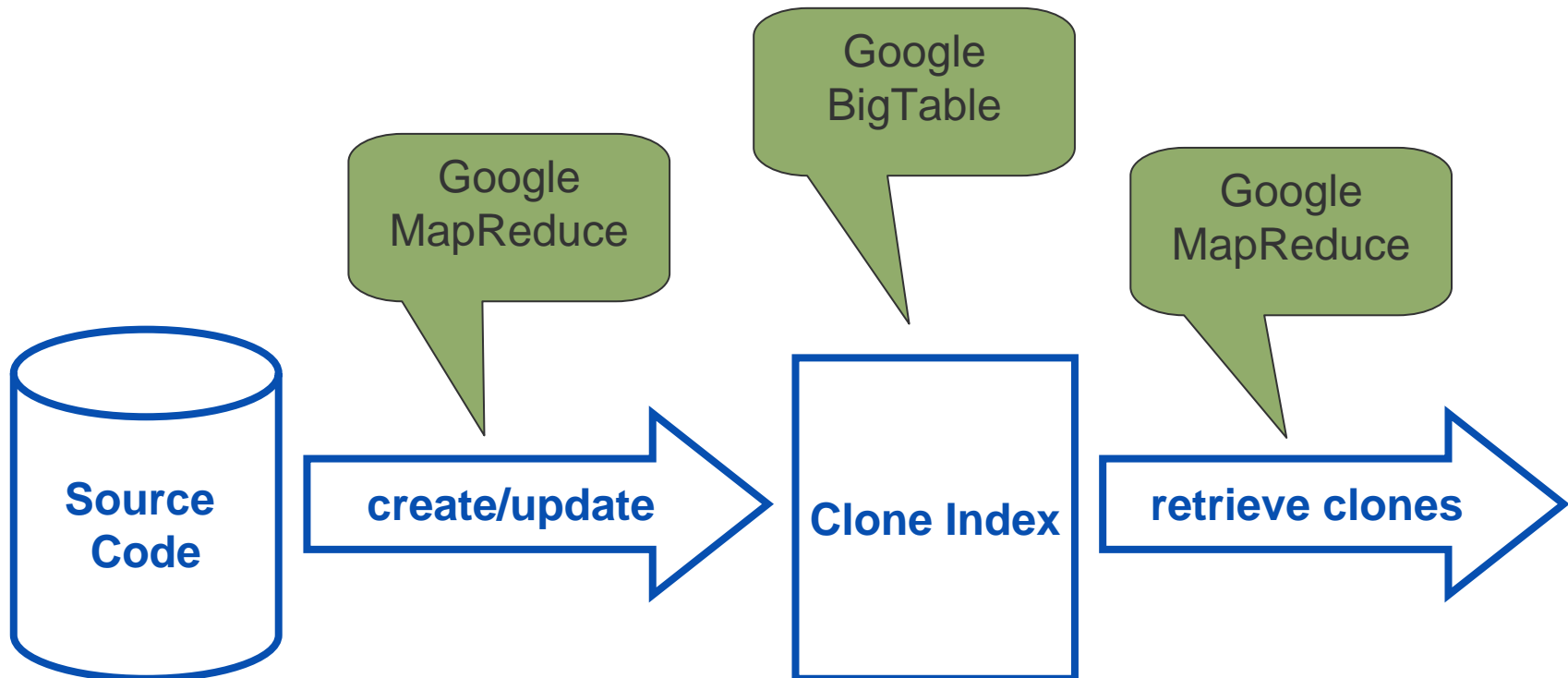
setScaleAtNullable(year);

rtter.Default.ToString(pre

# Clone Detection



# Index-Based Clone Detection

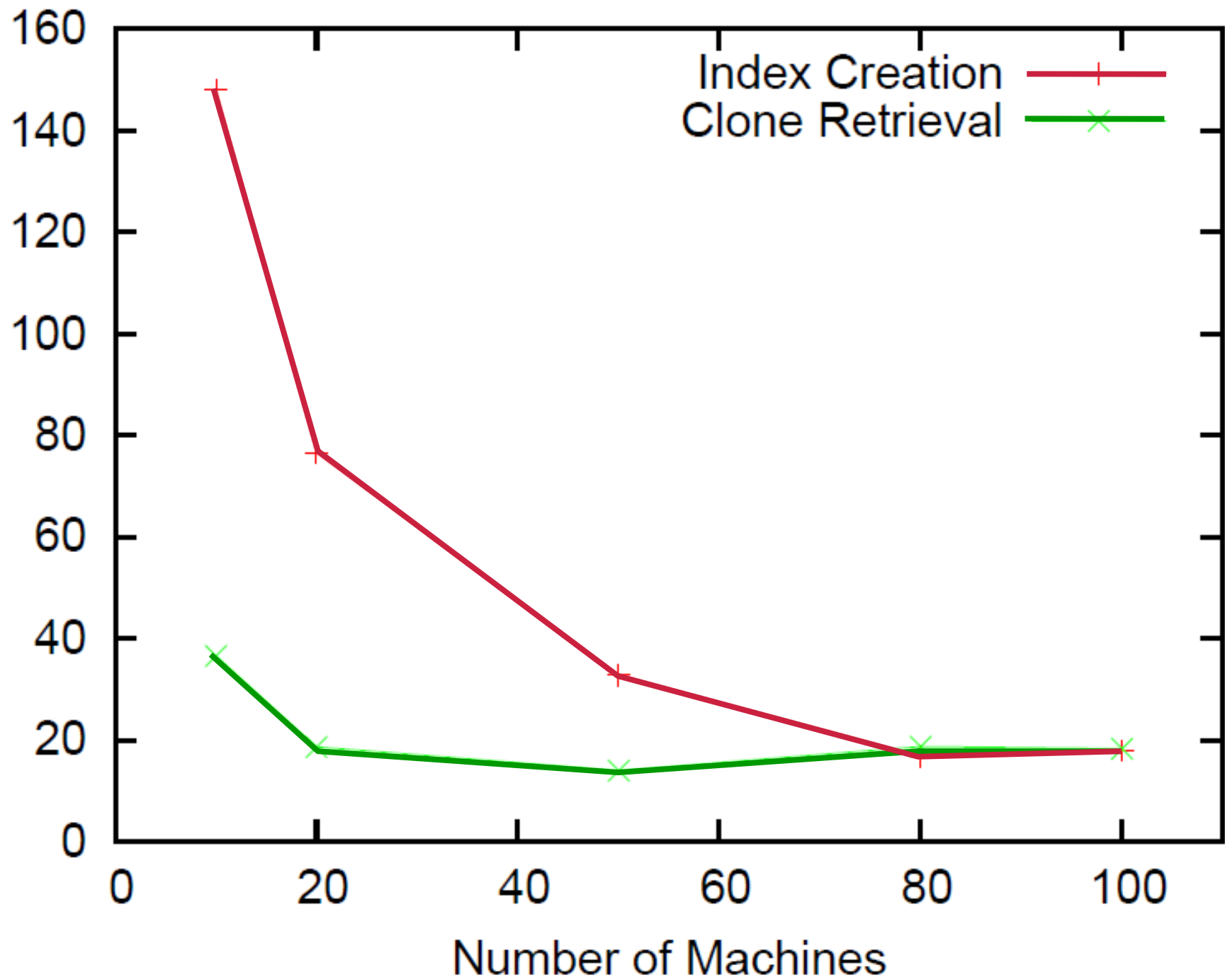


# Distributed Dection

## Google Infrastructure:

- **MapReduce** for distributing computing tasks
- **Bigtable** for storing index
- **73.2 MLOC** of Open Source code (C/C++/Java)\*
  
- Procedure:
- Determine running time on 10, 20, 50, 80, 100 machines

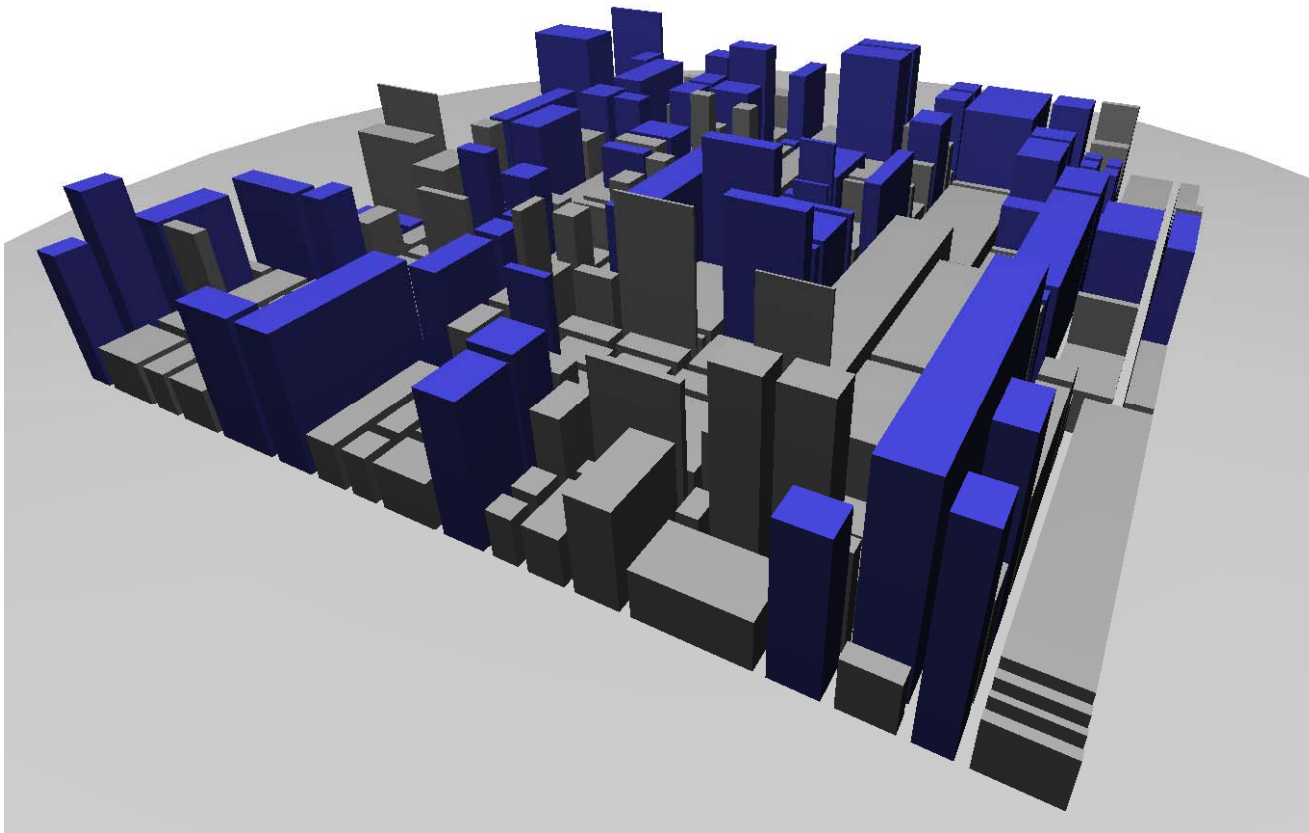
Execution Time in Minutes



# Ultra-large Scale Clone Detection

- Code base
  - 120 million C/C++ files
  - 2,915,947,163 Lines of code
  - Open source code from the index of Google Code Search
  
- Relevant use case: copyright infringement analysis
  - ⇒ Reporting all clones is useless
  - ⇒ Only the index creation was evaluated
  
- Index creation on 1000 machines took
  - less than 7 hours
  - ⇒ real scalability

# Visualization



## ConQAT in Industry

- Development support: output integrated into project dashboards.
- Monitoring of SW developed by subcontractors.
- Quality control of software developed at multiple globally distributed sites.
- Clone Detection on Matlab/Simulink.
- Part of quality assurance criteria at quality gates.
- Used during one-shot quality assessments.
- Applied for continuous control of C++ & PL/SQL Code



## Summary

ConQAT is about

- **integration** of diverse analyses, allowing
- **tailoring** to project specific questions for **continuous** quality assessment

ConQAT has been released under Apache License 2.0

<http://www.conqat.org>

Commercial support for ConQAT:

Contact: [deissenb@in.tum.de](mailto:deissenb@in.tum.de)

