{ COVER-REST }

Improved coverage analysis for LibreOffice's CI

Linus	linus@thebehrens.net
Thorsten	thb@libreoffice.org
Urs Svante	svante.schubert@libreoffice.org

A Prototype Fund project

{ COVER-REST }

The Prototype Fund is run by Open Knowledge Foundation Germany, and funded via Bundesministerium für Bildung und Forschung (BMBF).

OPEN KNOWLEDGE FOUNDATION DEUTSCHLAND

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

The Mission

for LibreOffice’s continuous integration:

0. develop the glue to integrate many different data providers
1. get new, shiny tools!
2. create incentives for developers and QA “to do the right thing”
3. provide automated means to locate “features” in LibreOffice

Our Team

Mr. C++ & DevOps
Thorsten Behrens

Mr. Python & UX
Linus Behrens

Mr. Java & Graph
Urs Svante Schubert

Cover-Rest project page:
https://cover-rest.gitlab.io/

Glue code for CI tools

Simplify n:m:o Problem

Project Stretch Goal: Improve (Coverage) Tool Modularity

n: Programming Languages

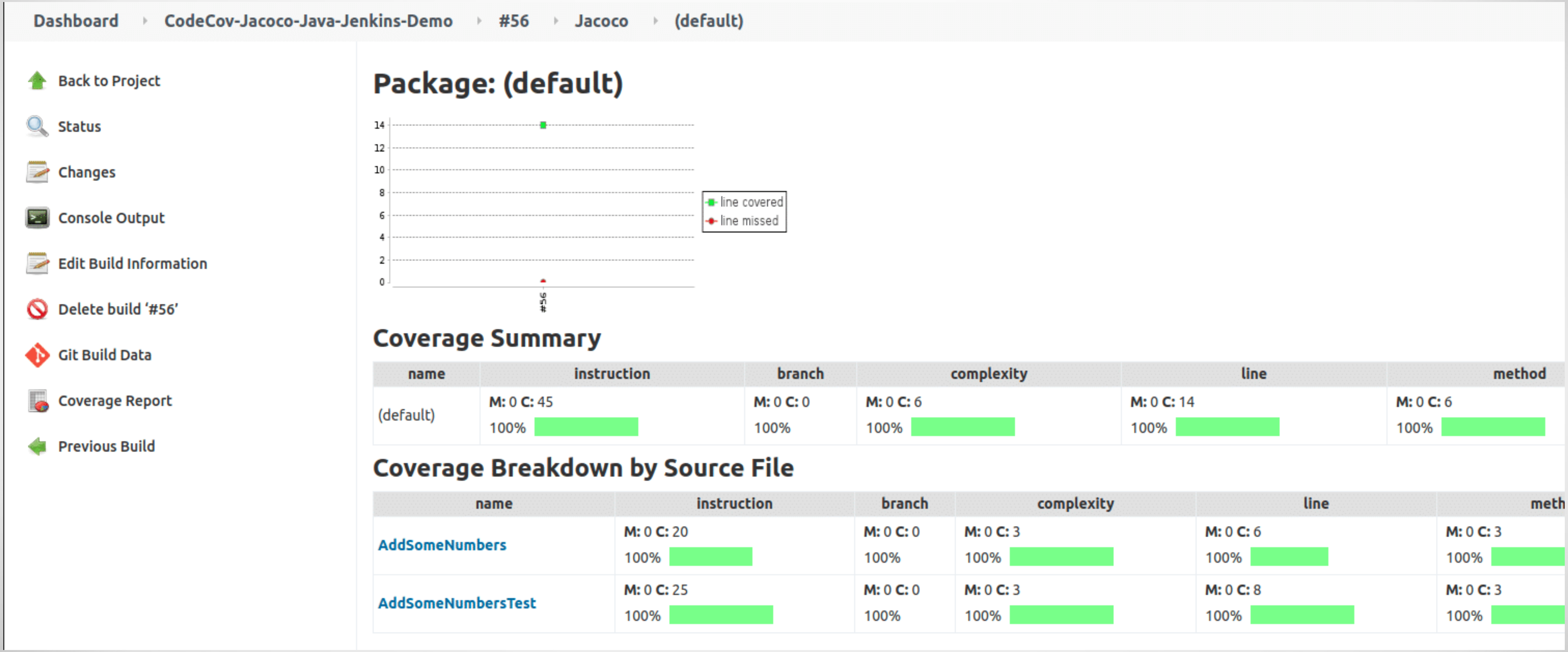
each Language has

m: Coverage Tools

which might need integration in **o:** CI systems

New, shiny tools!

Coverage – Jacoco Demo



Package: (default)

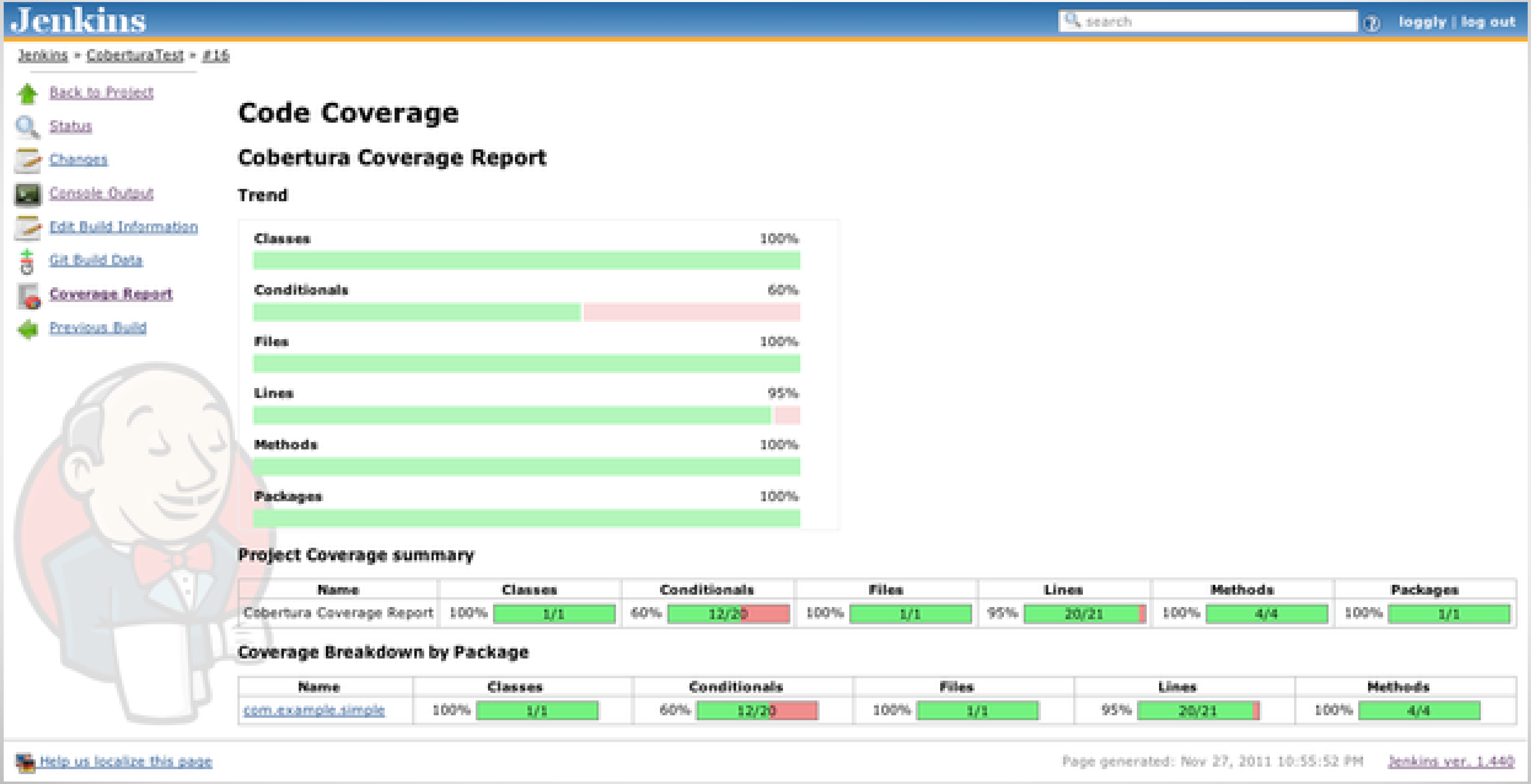
Coverage Summary

name	instruction	branch	complexity	line	method
(default)	**M:** 0 **C:** 45 100%	**M:** 0 **C:** 0 100%	**M:** 0 **C:** 6 100%	**M:** 0 **C:** 14 100%	**M:** 0 **C:** 6 100%

Coverage Breakdown by Source File

name	instruction	branch	complexity	line	meth
AddSomeNumbers	**M:** 0 **C:** 20 100%	**M:** 0 **C:** 0 100%	**M:** 0 **C:** 3 100%	**M:** 0 **C:** 6 100%	**M:** 0 **C:** 3 100%
AddSomeNumbersTest	**M:** 0 **C:** 25 100%	**M:** 0 **C:** 0 100%	**M:** 0 **C:** 3 100%	**M:** 0 **C:** 8 100%	**M:** 0 **C:** 3 100%

Coverage – Cobertura Demo



Code Coverage

Cobertura Coverage Report

Trend

Classes 100%

Conditionals 60%

Files 100%

Lines 95%

Methods 100%

Packages 100%

Project Coverage summary

Name	Classes		Conditionals		Files		Lines		Methods		Packages	
Cobertura Coverage Report	100%	1/1	60%	12/20	100%	1/1	95%	20/21	100%	4/4	100%	1/1

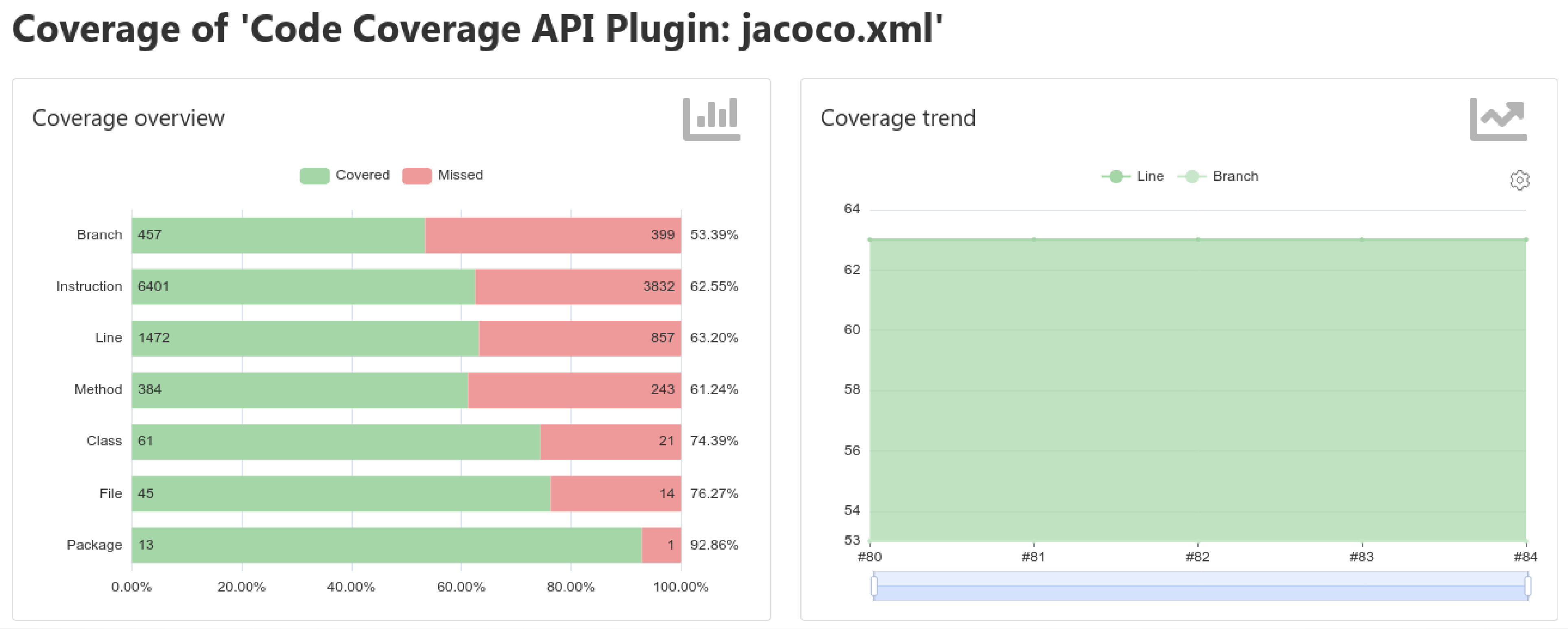
Coverage Breakdown by Package

Name	Classes		Conditionals		Files		Lines		Methods	
com.example.simple	100%	1/1	60%	12/20	100%	1/1	95%	20/21	100%	4/4

Help us localize this page

Page generated: Nov 27, 2011 10:55:52 PM Jenkins ver. 1.440

Coverage – API plugin using jacoco.xml

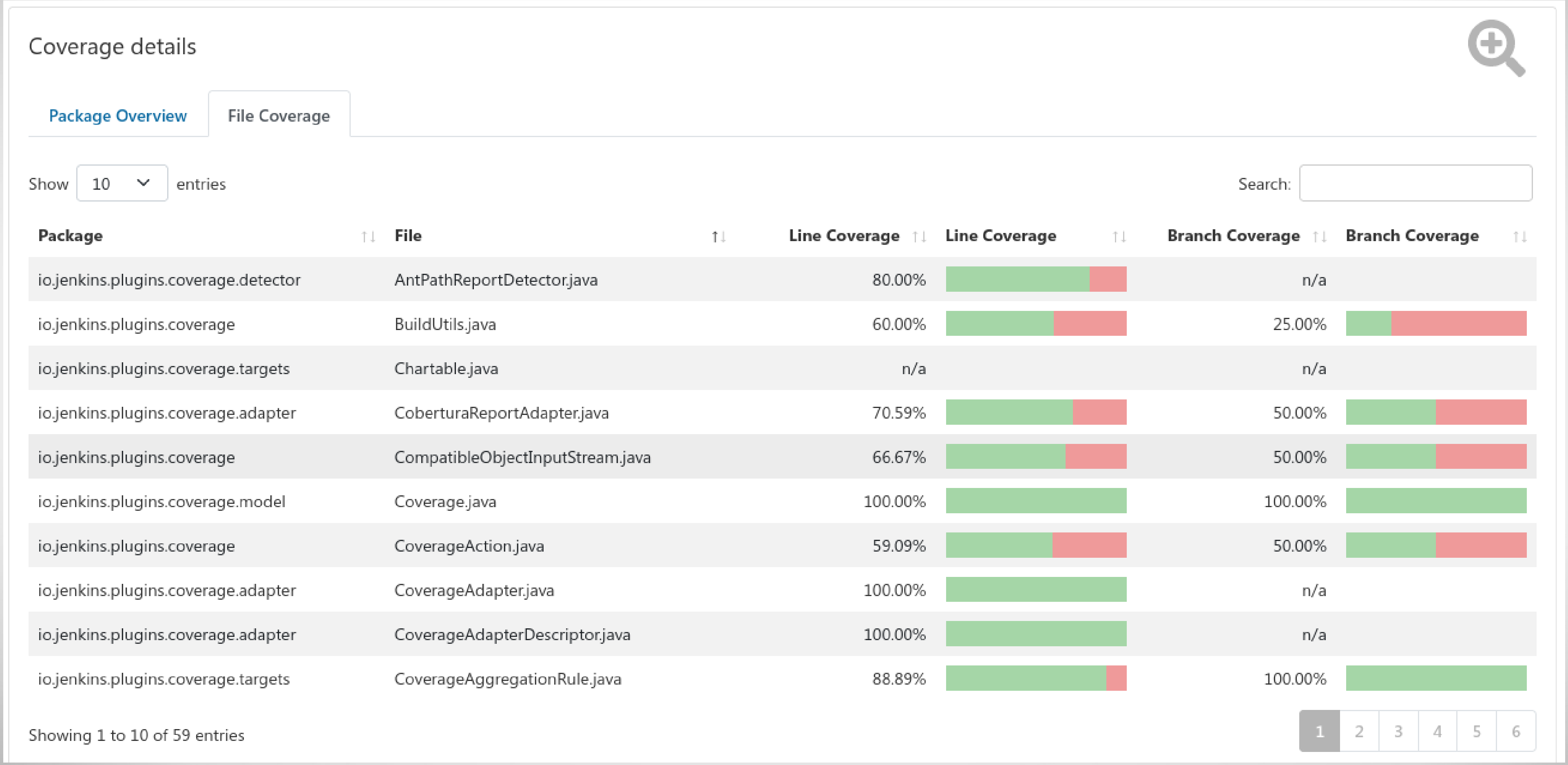


Coverage of 'Code Coverage API Plugin: jacoco.xml'

Coverage overview

Coverage trend

Coverage – API Plugin, drill-down list



Coverage details

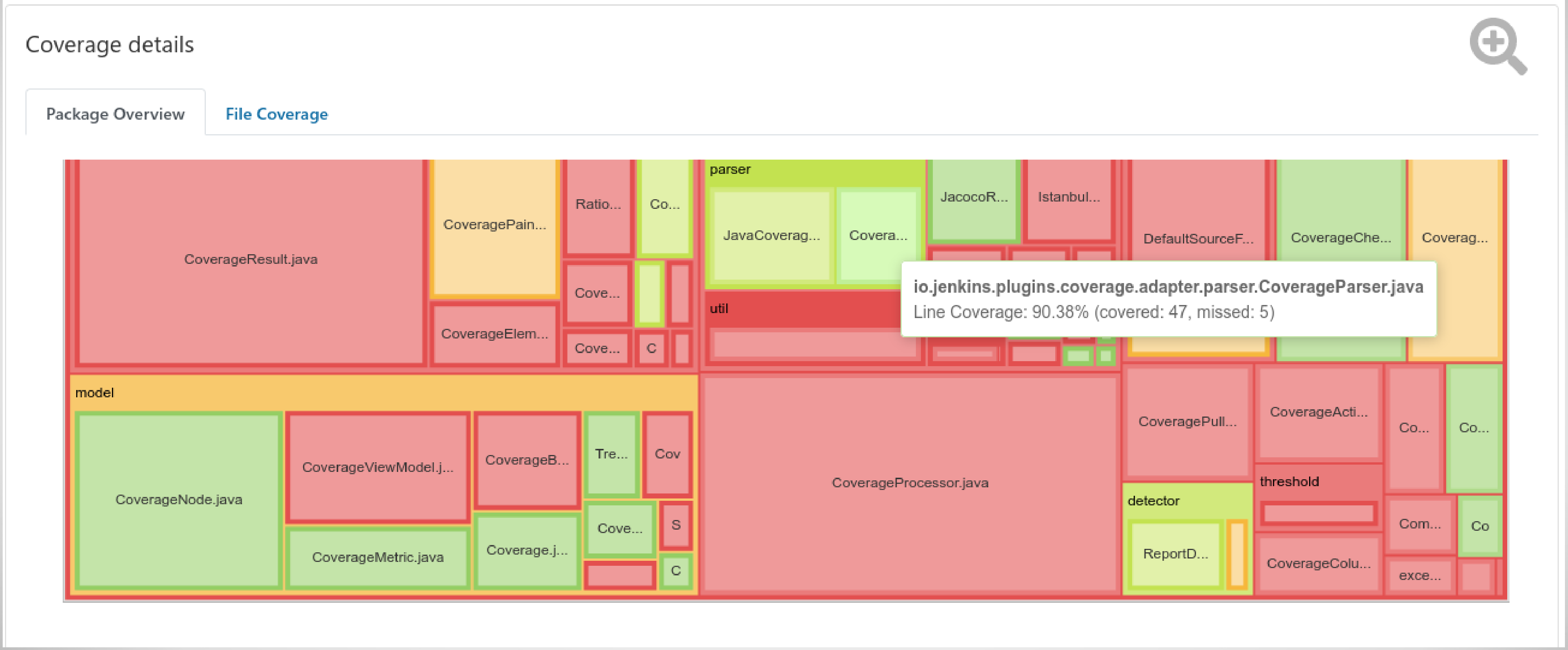
Show 10 entries

Search:

Package	File	Line Coverage	Line Coverage	Branch Coverage	Branch Coverage
io.jenkins.plugins.coverage.detector	AntPathReportDetector.java	80.00%	🟩🟥	n/a	
io.jenkins.plugins.coverage	BuildUtils.java	60.00%	🟩🟥	25.00%	🟩🟥
io.jenkins.plugins.coverage.targets	Chartable.java	n/a		n/a	
io.jenkins.plugins.coverage.adapter	CoberturaReportAdapter.java	70.59%	🟩🟥	50.00%	🟩🟥
io.jenkins.plugins.coverage	CompatibleObjectInputStream.java	66.67%	🟩🟥	50.00%	🟩🟥
io.jenkins.plugins.coverage.model	Coverage.java	100.00%	🟩	100.00%	🟩
io.jenkins.plugins.coverage	CoverageAction.java	59.09%	🟩🟥	50.00%	🟩🟥
io.jenkins.plugins.coverage.adapter	CoverageAdapter.java	100.00%	🟩	n/a	
io.jenkins.plugins.coverage.adapter	CoverageAdapterDescriptor.java	100.00%	🟩	n/a	
io.jenkins.plugins.coverage.targets	CoverageAggregationRule.java	88.89%	🟩🟥	100.00%	🟩

Showing 1 to 10 of 59 entries

Coverage – API Plugin, drill-down map

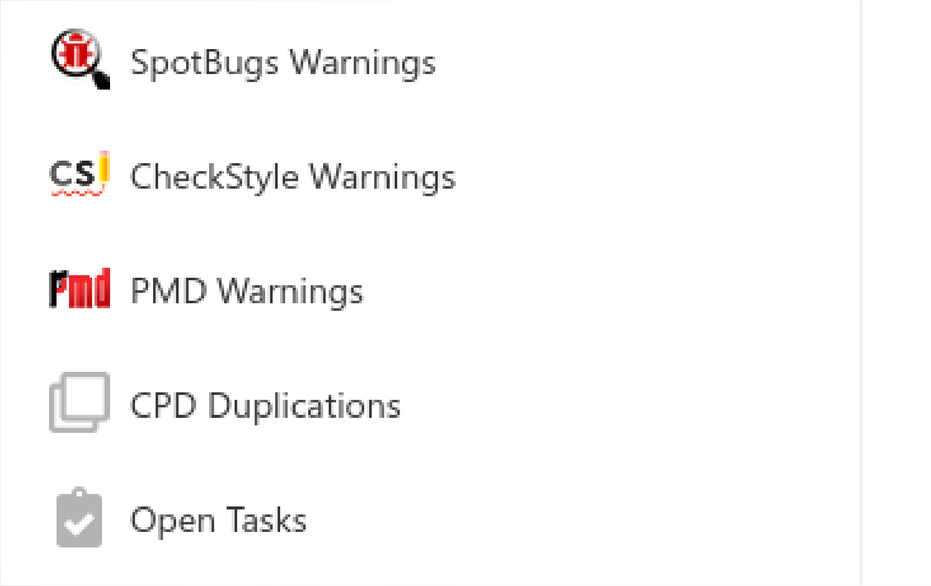


Coverage details

io.jenkins.plugins.coverage.adapter.parser.CoverageParser.java

Line Coverage: 90.38% (covered: 47, missed: 5)

More tools from the Jenkins ecosystem



SpotBugs Warnings

CheckStyle Warnings

PMD Warnings

CPD Duplications

Open Tasks

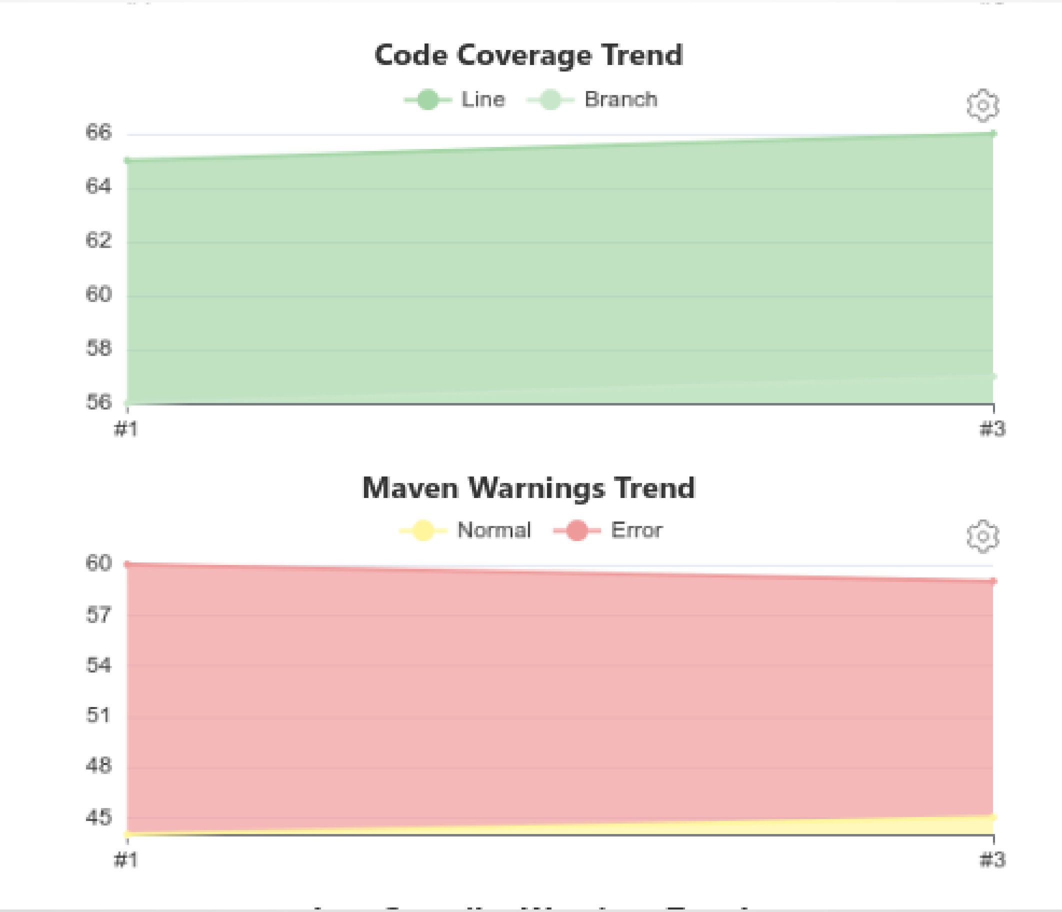
Jenkins has a host of plugins, and an active community (quite some software engineering research groups there too):

- Jacoco
- Cobertura
- Forensics-api
- Llvm-cov
- Opencover
- ...

Create incentives

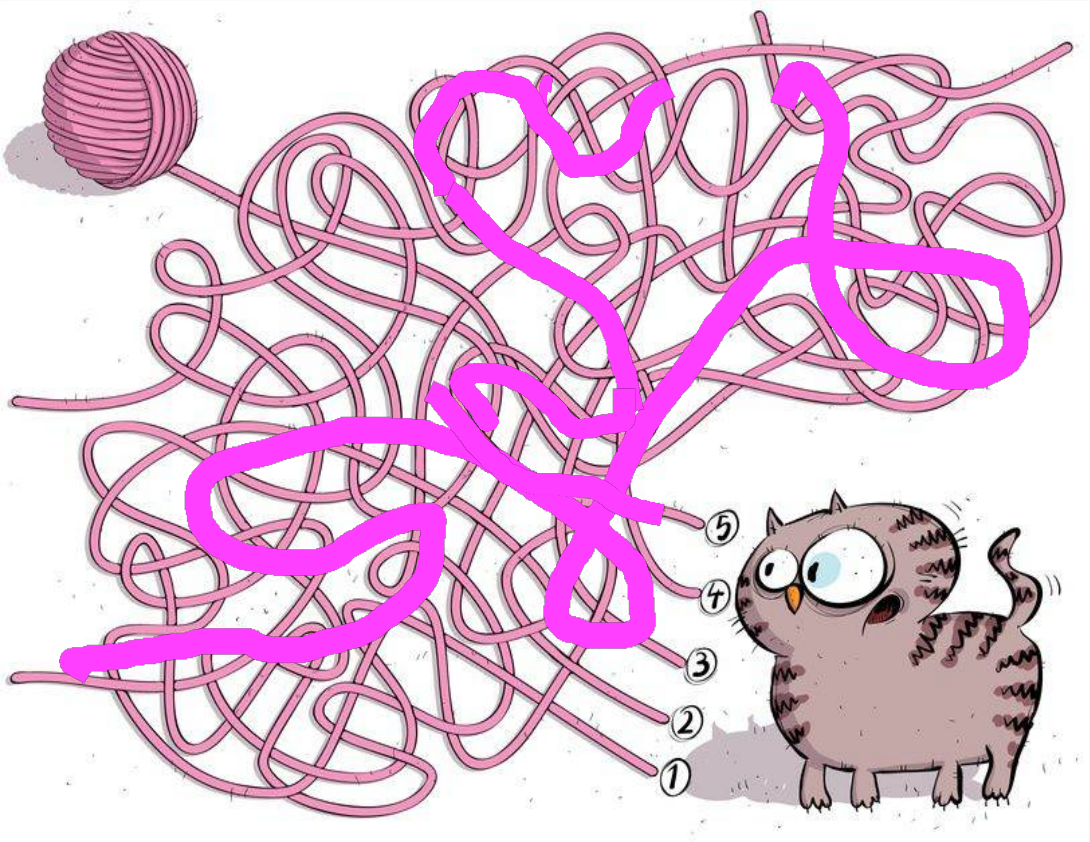
1. automate what is possible for reviews
2. nudge developers by suggesting changes
3. create incentives for developers – e.g. have nice metrics right inside patch review

Metrics visible at patch

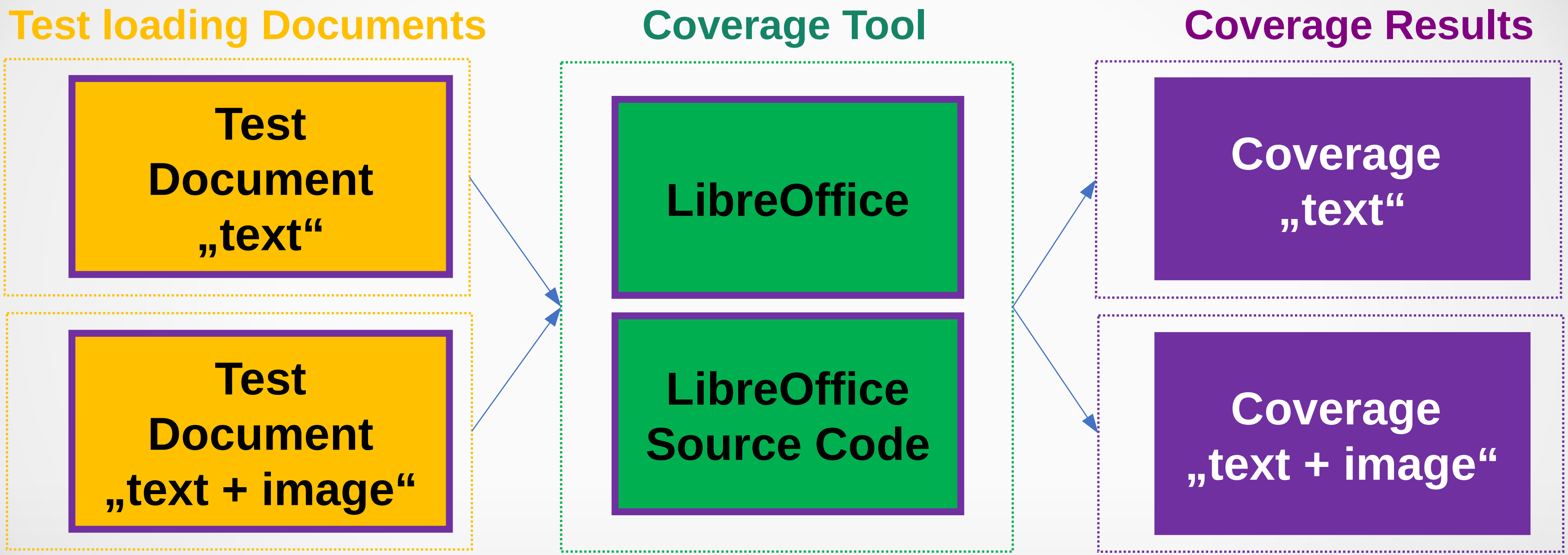


Feature map – “code cognita”

Determine control flow by “feature”



“Feature” load test in LibreOffice



Compute location of unique features

Subtracting of Coverages (DIFF)

Coverage **„text + image“**

-

Coverage **„text“**

= **Lines solely used by „image“ feature!**

Compute location of unique features

Subtracting of Coverages (DIFF)

Coverage **„text + image“**

$-$

Coverage **„text“**

= **Lines solely used by „image“ feature!**

Develop a “code cognita” map

Questions? Answers! :)